



D. LUIS SANTIAGO GARRIDO BULLÓN

con D.N.I.: 1.495.533

A U T O R I Z A :

Que su tesis doctoral con el título:
"IDENTIFICACIÓN, ESTIMACIÓN Y CONTROL
DE SISTEMAS NO LINEALES MEDIANTE RG0"
pueda ser utilizada para fines de investigación por
parte de la Universidad Carlos III de Madrid.

Leganés, 18 de febrero de 2000.

A handwritten signature in black ink, which appears to read 'Luis Santiago Garrido Bullón', is written over a horizontal line.

Fdo.: Luis Santiago Garrido Bullón



UNIVERSIDAD CARLOS III DE MADRID

Departamento de Ingeniería Eléctrica, Electrónica y Automática

TESIS DOCTORAL

Identificación, Estimación y Control de Sistemas No-lineales mediante RGO

Autor:
Santiago Garrido

Directores:
Dr. Luis Moreno
Dr. Carlos Balaguer

Leganés, 1999

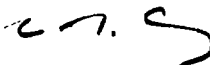
TESIS DOCTORAL


IDENTIFICACIÓN, ESTIMACIÓN Y CONTROL DE SISTEMAS NO- LINEALES MEDIANTE RGO


Autor: **Santiago Garrido Bullón**

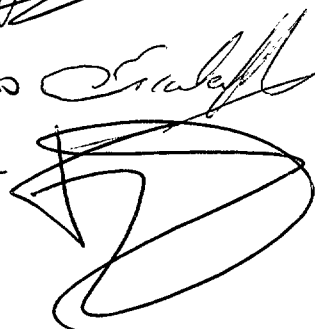
Directores: **Dr. Luis E. Moreno Lorente**
Dr. Carlos Balaguer Bernaldo de Quirós

Tribunal Calificador:

Presidente: **HIGUEL A. SAUCHS SANCHEZ CASSINERO** 

Vocales: **CÉSAR DE PEDRO MORDGA** 

GUSTIN JIMENEZ AVELLO 

DAVID DE LA ESCALERA MENDO
Vocal Secretario: **FRANCISCO MARCELLON ESPANOL** 

Calificación:

Leganés, 18 de FEBRERO de 2.000

A M^a José, Clara y Bruno,
por todo el tiempo que no
les he podido dedicar.



AGRADECIMIENTOS

Quiero dar las gracias en primer lugar a mis directores de tesis, D. Luis Moreno y D. Carlos Balaguer, por su inapreciable ayuda y buenos consejos, no sólo con el tema de la tesis, y que ahora culminan con la presentación de este trabajo.

A todos mis compañeros en el Departamento, por su desinteresada ayuda en todos los problemas que han ido surgiendo a lo largo de estos años, y por los buenos ratos que hemos pasado entre tanto.

En un aspecto más personal, debo dar las gracias a mi familia por aguantar y animar durante tanto tiempo a alguien que hace cosas tan raras como ésta.

RESUMEN

La identificación de sistemas trata de la estimación de modelos de sistemas dinámicos a partir de los datos observados. La estimación trata de evaluar y diseñar los estimadores de estado. Ambos se suponen que operan en un entorno estocástico.

Esta tesis dirige su esfuerzo investigador principalmente hacia la mejora de la resolución de los problemas de identificación y estimación de estados de sistemas dinámicos no-lineales y el control adaptativo de los mismos.

Se presenta un nuevo método híbrido para la optimización de funciones no lineales y no diferenciables que varían con el tiempo sin la utilización de derivadas numéricas. Esto es especialmente importante por la presencia de ruido. Este método está basado en los algoritmos genéticos con una nueva técnica de búsqueda que hemos denominado Optimización Genética Restringida (RGO). Este método está especialmente indicado para su uso en la identificación para el control de sistemas dinámicos.

Este método de optimización es el marco unificador de la tesis y al ser un método básico permite su aplicación a multitud de problemas relacionados con la Teoría de Sistemas, que precisan de una optimización en línea, puesto que las funciones varían con el tiempo y sin el uso de derivadas numéricas ya que estos sistemas están expuestos a ruidos.

A partir de éste algoritmo, se presenta un método de altas prestaciones para la identificación de sistemas no lineales variables con el tiempo con modelos lineales y no lineales. Este método puede ser usado en línea y en bucle cerrado, por lo que se puede utilizar en control. Este método utiliza un algoritmo de identificación en línea que empieza calculando qué ARX se adapta mejor al sistema, lo que nos da los órdenes y el retardo del modelo. A continuación calcula un modelo ARMAX que sirve como semilla para inicializar el RGO y crear un modelo NARMAX.

El algoritmo RGO permite describir un nuevo estimador no lineal para el filtrado de sistemas con procesos y modelos de observación no lineales, basados en la optimización con RGO. Los resultados de simulación se usan para comparar el rendimiento de este método con EKF (Extended Kalman Filter), IEKF (Iterated Extended Kalman Filter), SNF (Second-order Non-linear Filter), SIF (Single-

stage Iterated Filter) y MSF (Montecarlo Simulation Filter) en presencia de diferentes niveles de ruido. Asimismo se compara con las Redes Neuronales de tipo Backpropagation.

Cuando se aplica a la identificación en el espacio de estados se obtiene un nuevo método de identificación que comienza calculando un ARX y posteriormente utiliza RGO para mejorar la identificación previa. Este método está basado en el modelo de parametrización completa y las realizaciones equilibradas. De esta manera se consiguen realizaciones de baja sensibilidad y que evitan los problemas de estructura de las realizaciones canónicas.

Se presentan dos aplicaciones de estos métodos. La primera es el control predictivo con RGO del Sistema MIMO de Rotores Gemelos (Twin Rotor MIMO System - TRMS), que es un laboratorio diseñado expresamente para realizar experimentos de control. En ciertos aspectos su comportamiento recuerda al de un helicóptero. Desde el punto de vista de control ejemplifica un sistema de orden alto con acoplamientos cruzados muy significativos.

La segunda aplicación es la localización de robots mediante la información de los distintos sensores. Para fusionar toda esta información y así poder corregir la posición y orientación del vehículo es necesario un algoritmo que en este caso es una extensión del filtro de Kalman usando RGO.

ABSTRACT

The system identification deals with the problem of estimating models of dynamical systems from observed data. The estimation tries to evaluate and to design state estimators. The two of them are supposed to operate in a stochastic environment.

In this thesis, It has been tried to improve the methods of identification and state estimation of non-linear dynamical systems and their adaptive control.

A new optimization hybrid method of non-linear and non-differentiable, time-varying functions without using numerical derivatives is presented. This is important because of noise. This method based on Genetic Algorithms introduces a new technique called Restricted Genetic Optimization (RGO).

This optimization method unifies the thesis and due to the fact that it is a basic method, it can be applied to a lot of problems related with non-differentiable and time-varying functions.

Based on this algorithm, a high performance method for the identification of non-linear, time-varying systems with linear and non-linear models, is presented. This method can be used on-line and in a closed loop. For this reason, it is well adapted to control. This method uses an on line identification algorithm that begins by calculating what ARX is the best adapted to the system. This way the order and the delay of the system are known. Then, an ARMAX that is used as a seed to start the RGO and to create a NARMAX model, is calculated.

The RGO algorithm can describe a new non-linear estimator for filtering of systems with non-linear processes and observation models based on the RGO optimization. The simulation results are used to compare the performance of this method with EKF (Extended Kalman Filter), IEKF (Iterated Extended Kalman Filter), SNF (Second-order Non-linear Filter), SIF (Single-stage Iterated Filter) y MSF (Montecarlo Simulation Filter) with different levels of noise.

When this method is applied to the state space identification a new method is obtained. This method begins by calculating an ARX and then uses RGO in order to improve the previous identification. This method is based on the full parametrization and balanced realizations. This way low sensitivity realizations are obtained and the structural issues of multivariable canonical parametrizations

are circumvented.

Two applications of this method are considered. The first application is the predictive control with RGO of the Twin Rotor MIMO System (TRMS), that is a laboratory set-up designed for control experiments. In certain aspects, its behaviour resembles that of a helicopter. From the control point of view, it exemplifies a high order non-linear system with significant cross-couplings.

The second one is the robot localization based on different kind of sensor information. To fuse all the different information, an algorithm is necessary. In this case, it has been used an extension of the Kalman algorithm with RGO.

ÍNDICE GENERAL

Agradecimientos	v
Resumen	vii
Abstract	ix
Introducción	xxiii
1. Fundamentos y Estado del Arte.	1
1.1 Identificación	1
1.1.1 Modelos de Sistemas Dinámicos	2
1.1.2 Modelos de Entrada-Salida	3
1.1.3 Modelos Paramétricos Lineales	4
1.1.4 Modelos Paramétricos No-lineales	5
1.1.5 Modelos No-Paramétricos, Redes Neuronales	8
1.1.6 Identificación con Algoritmos Genéticos	15
1.1.7 Identificación en Bucle Cerrado	16
1.2 Estimación de Estados	18
1.2.1 Modelos en el Espacio de Estados	18
1.2.2 La estimación lineal en sistemas dinámicos - El Filtro de Kalman	19
1.2.3 El caso no lineal: Filtros No-lineales	20
1.2.4 Filtro Extendido de Kalman (EKF) y Filtro No-lineal de Segundo Orden (SNF)	22
1.2.5 Filtro extendido Iterado de Kalman (IEKF)	22
1.3 Validación de Modelos	23
1.4 Control Adaptativo	25
2. Conceptos Básicos de los A.G.	27
2.1 Algoritmos Genéticos	27
2.1.1 Algoritmo Genético Simple	30

2.1.2	Mecanismos Avanzados	35
2.2	Algoritmos Genéticos con parámetros continuos	41
3.	Estudio matemático de los A.G.	43
3.1	Introducción a los procesos de Markov	43
3.2	Clasificación de estados	45
3.3	Algoritmos Genéticos y Cadenas de Markov	46
3.4	Convergencia y Comportamiento Asintótico	49
4.	Método de Optimización Genética Restringida (RGO)	53
4.1	Introducción	53
4.2	Características de los Algoritmos Genéticos	54
4.3	Optimización Genética Restringida (RGO)	54
4.3.1	RGO	55
4.4	Algoritmo	57
4.5	Características de la RGO	59
5.	Identificación de sistemas no lineales	61
5.1	Introducción	61
5.2	Implementación del Algoritmo	62
5.3	Identificación multimodelo y multiparámetro de sistemas no lineales mediante modelos paramétricos lineales.	63
5.3.1	Identificación con modelos lineales	63
5.3.2	Identificación RGO para modelos lineales	64
5.4	Identificación multimodelo y multiparámetro	67
5.4.1	Modelos NARMAX	67
5.4.2	Identificación RGO con modelos no lineales	68
5.5	Restricciones	69
5.6	Resultados	70
5.7	Comparación con las Redes Neuronales	75
5.8	Conclusiones	79
6.	Estimación de Estados	81
6.1	Introducción	81
6.2	El Problema de Filtrado.	82
6.2.1	La estimación lineal en sistemas dinámicos - El Filtro de Kalman	83
6.3	Filtros No-lineales Tradicionales	86
6.3.1	Filtro Extendido de Kalman (EKF) y Filtro No-lineal de Segundo Orden (SNF)	86

6.3.2	Filtro extendido Iterado de Kalman (IEKF)	88
6.3.3	Filtro Iterado de una Etapa (Single-stage Iterated Filter (SIF))	89
6.3.4	Filtro con Simulación Montecarlo (MSF)	89
6.4	El nuevo Filtro, RGO	90
6.5	Comparación de Filtros No-lineales.	92
6.6	Discusión	95
7.	Identificación en el Espacio de Estados	97
7.1	Modelos de Sistemas Lineales en el Espacio de Estados	97
7.2	Parametrización e Identificabilidad	98
7.3	Modelos en el Espacio de Estados Totalmente Parametrizados	99
7.4	Realizaciones Equilibradas (Balanced Realizations)	100
7.5	Identificación RGO en el Espacio de Estados	101
7.5.1	El Algoritmo	104
7.6	Resultados	105
7.7	Discusión	105
8.	Aplicaciones al Control Predictivo	107
8.1	Introducción al Control Predictivo Basado en Modelo	107
8.1.1	Estrategias de Control Predictivo Basado en Modelo	108
8.1.2	La función objetivo	109
8.1.3	Trayectoria de referencia	110
8.2	Control Predictivo GPC.	110
8.3	Ejemplo.	111
8.4	Control Predictivo RGO.	112
8.4.1	Ventajas de utilizar RGO en un GPC.	115
8.5	Control Predictivo PID-RGO	116
8.5.1	Introducción	116
8.5.2	Identificación mediante Modelos Paramétricos Lineales	117
8.5.3	Control Predictivo PID-RGO	119
8.5.4	Simulación del control adaptativo PID-RGO.	120
8.6	Análisis de resultados	120
9.	Aplicaciones I: TRMS	123
9.1	Introducción	123
9.2	Modelado y parámetros	125
9.3	Identificación del TRMS	127
9.4	Control PID del TRMS	128

9.4.1	Estabilización y seguimiento 1-DOF vertical	128
9.4.2	Estabilización y seguimiento 1-DOF horizontal	130
9.4.3	Estabilización y seguimiento 2-DOF.	130
9.4.4	Controlador PID con acoplamiento cruzado	131
9.5	Control Predictivo mediante PID y RGO	133
9.6	Control predictivo del TRMS	135
9.7	Control en el Espacio de Estados del TRMS	139
9.8	Comparación de los distintos métodos	140
10.	Aplicaciones II: Robot Móvil	143
10.1	El Filtro de Kalman Extendido Iterado	143
10.2	Localización del Robot Móvil utilizando un Filtro RGO No-lineal	143
10.2.1	Consideraciones de diseño	145
10.2.2	La estructura genética	146
10.2.3	La función de salud	146
10.3	Resultados experimentales	148
11.	Conclusiones y Trabajo Futuro.	151
11.1	Conclusiones	151
11.1.1	Resumen y conclusiones	151
11.2	Aportaciones principales	151
11.3	Trabajo Futuro	152

ÍNDICE DE TABLAS

2.1	Fenotipos y salud asociada.	31
2.2	Número de copias esperado.	32
2.3	Cruce	33
2.4	Cruce	34
2.5	Mapa de dominación.	38
3.1	El número de estados N como función de l y P	47
3.2	La matriz Z para $l = 2$ y $P = 2$	48
5.1	Comparación de los métodos de identificación de sistemas.	74
5.2	Comparación de los métodos de identificación: RGO y Redes Neuronales.	77
6.1	Comparación de BIAS para el nivel de ruido 0.1.	94
6.2	Comparación de RMSE para el nivel de ruido 0.1.	94
6.3	Comparación de BIAS para el nivel de ruido 0.5.	95
6.4	Comparación de RMSE para el nivel de ruido 0.5.	95
6.5	Comparación de BIAS para el nivel de ruido 1.0.	95
6.6	Comparación de RMSE para el nivel de ruido 1.0.	96
7.1	Errores Cuadráticos Relativos	105
9.1	Notación usada en la figura 9.2	126
9.2	Resultados de la comparación entre los distintos métodos de control del TRMS.	142
10.1	Localización en posición con y sin RGO (1). Error en mm.	148
10.2	Localización angular con y sin RGO (1). Error en grados.	149
10.3	Localización en posición con y sin RGO (2). Error en mm.	149
10.4	Localización angular con y sin RGO (2). Error en grados.	149

ÍNDICE DE FIGURAS

1.1	Red Feedforward con una capa oculta y una salida.	9
1.2	Red Feedforward con dos capas ocultas.	10
1.3	Red Recurrente. q^{-1} retarda la señal un tiempo de muestreo.	10
1.4	Identificación Serie-Paralelo	11
1.5	Identificación Paralela	12
1.6	Red con realimentación de la Capa Oculta.	13
1.7	Red con realimentación de la Capa de Salida.	14
1.8	Sistema en bucle cerrado.	17
1.9	Diagrama de flujo del filtro de Kalman.	21
1.10	Sistema adaptativo con identificación de modelo (MIAS)	25
1.11	Sistema adaptativo con modelo de referencia (MRAS)	26
2.1	Algoritmos Genéticos.	27
2.2	Búsqueda Genética.	28
2.3	Terminología de los Algoritmos Genéticos.	29
2.4	Ciclo de los Algoritmos Genéticos.	30
2.5	¿Cómo funcionan los Algoritmos Genéticos?	30
2.6	Métodos de selección.	31
2.7	Función de salud.	32
2.8	Cruce.	33
2.9	Mutación.	34
4.1	Mecanismo de búsqueda de la Optimización Genética Restringida.	56
4.2	Mecanismo de seguimiento de la Optimización Genética Restringida.	58
4.3	Diagrama de flujo de la Optimización Genética Restringida RGO	60
5.1	Diagrama de flujo de la RGO.	66
5.2	Mecanismo de cruce	68
5.3	Mecanismo de mutación	68
5.4	Montaje de la Planta 2.	71

5.5	Fenómeno de blow-up al intentar identificar la planta 1 con el bloque ARMAX de Simulink. Este efecto se debe a que se correlan los errores $\varepsilon(t)$ y $\eta(t)$ con la entrada y hace imposible la identificación de este sistema mediante este método.	72
5.6	Resultados de la identificación de la planta 1 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido. .	72
5.7	Resultados de la identificación de la planta 2 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido. .	73
5.8	Resultados de la identificación de la planta 3 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido. .	73
5.9	Resultados de la identificación de la planta 4 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido. .	74
5.10	Estructura de la Red Multicapa utilizada en la comparación. . . .	75
5.11	Resultados de la identificación de la planta 1. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.	77
5.12	Resultados de la identificación de la planta 2. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.	78
5.13	Resultados de la identificación de la planta 3. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.	78
5.14	Resultados de la identificación de la planta 4. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.	79
6.1	Diagrama de flujo del filtro de Kalman.	85
6.2	Diagrama de flujo de la Optimización Genética Restringida aplicada a la estimación de estados.	91
6.3	Ejecuciones típicas para el modelo 5 con EKF y SRGOF para nivel de ruido 1.0.	93
7.1	Twin Rotor MIMO System (TRMS).	102
7.2	Resultados de identificación de la Planta con el método ARX. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error . .	103

7.3	Resultados de identificación de la Planta con el método Gauss-Newton. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error	103
7.4	Resultados de identificación de la Planta con el método RGO. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error . .	104
8.1	La metodología del control predictivo basado en modelos.	108
8.2	Estructura básica del control predictivo.	109
8.3	Distintas trayectorias de referencia.	110
8.4	Resultados de la identificación de los parámetros l_{y1} , l_{y2} y l_{r1} del control predictivo RGO.	114
8.5	Esquema para la identificación de los parámetros de la ley de control predictivo.	115
8.6	Resultados de la identificación de los parámetros l_{y1} , l_{y2} y l_{r1} del control predictivo RGO. El tiempo 10 corresponde al llenado del buffer.	115
8.7	Resultados de la identificación de los parámetros del sistema. . . .	116
8.8	Esquema general del control adaptativo.	117
8.9	117
8.10	Diagrama Simulink para Control 1	121
8.11	Resultados del Control Predictivo con RGO para $\lambda = 0.12$	121
8.12	Resultados del Control Predictivo con RGO para $\lambda = 0.16$	122
8.13	Resultados del Control Predictivo con RGO para $\lambda = 0.20$	122
9.1	Twin Rotor MIMO System (TRMS).	123
9.2	Twin Rotor MIMO System (TRMS).	124
9.3	Diagrama de bloques del modelo TRMS.	125
9.4	Respuesta del rotor principal del TRMS a una onda cuadrada. . . .	128
9.5	Control del sistema 1-DOF (estabilización y seguimiento en vertical). Arriba: salida vs. señal de referencia. Abajo: error cometido.	128
9.6	Diagrama de bloques del sistema 1-DOF (plano vertical).	129
9.7	Control del sistema 1-DOF (estabilización y seguimiento en vertical). Arriba: salida del sistema vs. señal de referencia. Abajo: error cometido.	129
9.8	Control del sistema 1-DOF (estabilización y seguimiento horizontal).	130
9.9	Diagrama de bloques del sistema 1-DOF (plano horizontal). . . .	130
9.10	Control del sistema 1-DOF (estabilización y seguimiento en horizontal). Arriba: salida del sistema vs. señal de referencia. Abajo: error cometido.	131

9.11	Control del sistema 2-DOF.	131
9.12	Experimento con el controlador de estabilización y seguimiento 2-DOF PID. Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo : error cometido	132
9.13	Experimento con el controlador de estabilización y seguimiento 2-DOF PID. Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo : error cometido	132
9.14	Diagrama de bloques del sistema de control 2-DOF PID de acoplamiento cruzado.	133
9.15	Experimento con el controlador de estabilización y seguimiento 2-DOF PID con acoplamiento cruzado. Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo : error cometido . . .	133
9.16	Experimento con el controlador de estabilización y seguimiento 2-DOF PID con acoplamiento cruzado. Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo : error cometido. .	134
9.17	Sistema adaptativo con identificación de modelo (MIAS)	135
9.18	Control del sistema con RGO-PID y RGO Predictivo.	136
9.19	Experimento con el controlador de estabilización y seguimiento: control predictivo-RGO en el plano vertical (con un grado de libertad). Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo: error cometido.	136
9.20	Experimento con el controlador de estabilización y seguimiento: PID-RGO (arriba) en el plano vertical (con un grado de libertad). Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo : error cometido.	137
9.21	Experimento con el controlador de estabilización: Predictivo-RGO con dos grados de libertad. Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo: error cometido.	137
9.22	Experimento con el controlador de estabilización: PID-RGO con dos grados de libertad. Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo: error cometido.	138
9.23	Experimento con el controlador de estabilización (señales de control: control predictivo y PID).	138
9.24	Control del sistema 2-DOF.	139
9.25	Experimento con el controlador de estabilización y seguimiento: PID-RGO en variables de estado. Arriba: salida del sistema vs. referencia en eje vertical. Abajo: salida del sistema vs. referencia en eje horizontal.	140

9.26 Experimento con el controlador de estabilización: diagrama de bloques.	141
---	-----



INTRODUCCIÓN

Ha habido un gran interés por parte de la comunidad de control en aplicar los algoritmos genéticos a problemas de Ingeniería de Control de Sistemas. Basados en modelos computacionales de la evolución biológica natural, los Algoritmos Genéticos pertenecen a la clase de los algoritmos evolutivos, que también incluyen la programación evolutiva, la evolución de estrategias y la programación genética. Comparado con los métodos tradicionales de búsqueda y optimización, tales como los basados en el cálculo o los basados en técnicas enumerativas, los algoritmos genéticos son una técnica robusta, que puede ser utilizada de forma global o semi-local y que se puede aplicar generalmente de forma más directa a situaciones que tienen poco o ningún conocimiento acerca del proceso a ser controlado. Para el ingeniero de control los algoritmos genéticos y la programación evolutiva, en general, presentan oportunidades de plantearse problemas que no permiten una solución eficiente a través de la aplicación de técnicas convencionales. En los últimos años, los algoritmos genéticos han sido aplicados a muchos temas relacionados con la Ingeniería de Control, incluyendo la optimización paramétrica y combinatoria y el control adaptativo.

El diseño de controladores no-lineales es un problema muy importante en Ingeniería de Control, aunque difícil. Hoy en día, la mayoría de los controladores industriales son lineales puros debido a las dificultades encontradas en el problema de diseño de controladores no lineales. Para construir un controlador no-lineal hemos de hacer suposiciones válidas acerca de la estructura del controlador y del sistema a controlar. Estas suposiciones nos llevan frecuentemente al desarrollo de controladores no suficientemente robustos puesto que los efectos de la incertidumbre no han sido tenidos en cuenta a la hora de diseñar el controlador.

La identificación de sistemas de dinámica desconocida ha producido un campo de investigación llamado 'identificación de sistemas'. En identificación de sistemas definimos o describimos una estructura modelo que tiene un comportamiento similar al del sistema desconocido.

Objetivos de la tesis

El principal objetivo de la tesis es el desarrollo de nuevos métodos híbridos

de identificación y de estimación de estados, 'on line' de sistemas dinámicos no-lineales, basados en algoritmos genéticos. Asimismo, se han estudiado sus aplicaciones al control adaptativo de dichos sistemas.

En la literatura, el uso de los algoritmos genéticos como método de optimización se hace off-line y el tiempo utilizado es bastante grande, ya que usualmente los algoritmos genéticos se usan como un método de búsqueda global y en paralelo.

Hemos utilizado nuevos mecanismos para usarlos como lo hace la Naturaleza: On-line y de una manera local o semi-local.

Hemos demostrado que los algoritmos genéticos se pueden usar como un método rápido de optimización de funciones que varíen con el tiempo.

A éste nuevo método que hemos desarrollado lo hemos denominado Restricted Genetic Optimization (RGO) y simula el método del gradiente sin utilizar derivadas y puede ser utilizado cuando las señales tienen mucho ruido.

El método de búsqueda restringida consiste en hacer la búsqueda en un entorno del punto correspondiente al último modelo identificado y tomar el punto mejor adaptado como centro del entorno de búsqueda de la nueva generación.

Los individuos de cada generación representan los vectores de diferencia respecto al mejor punto de la generación anterior. El radio se toma proporcional a la incertidumbre.

Estructura de la Tesis

El contenido de la tesis se encuentra distribuido en diez capítulos. Inicialmente se presenta un capítulo dedicado a las distintas metodologías relacionadas con el contenido de la tesis.

El desarrollo de un nuevo método de optimización RGO (Restricted Genetic Optimization) para funciones que varíen con el tiempo. Este método ha probado ser muy útil para minimizar o maximizar funciones posiblemente no-lineales y no-diferenciables. Este método está especialmente indicado para su uso en la identificación para el control de sistemas dinámicos.

En el capítulo 6 se presenta un método de altas prestaciones para la identificación de sistemas no lineales variables con el tiempo con modelos lineales y no lineales mediante RGO.

El capítulo 7 trata de nuevo estimador no lineal para la estimación de los estados en sistemas con procesos y modelos de observación no lineales, basados en la optimización con RGO (Restricted Genetic Optimization). Los resultados de simulación se usan para comparar el rendimiento de este método con EKF (Extended Kalman Filter), IEKF (Iterated Extended Kalman Filter), SNF (Second-order Nonlinear Filter), SIF (Single-stage Iterated Filter) y MSF (Montecarlo Simula-

tion Filter) en presencia de diferentes niveles de ruido. Asimismo se compara con las Redes Neuronales de tipo Backpropagation.

En el capítulo 8 mostramos un nuevo método de identificación de sistemas dinámicos con modelos en el Espacio de Estados. Este método un algoritmo de identificación que comienza calculando un ARX y posteriormente utiliza RGO para mejorar la identificación previa.

Las aplicaciones de estos métodos al control se tratan en el capítulo 9 a partir del Sistema MIMO de Rotores Gemelos (Twin Rotor MIMO System - TRMS), que es un laboratorio diseñado expresamente para realizar experimentos de control. En ciertos aspectos su comportamiento recuerda al de un helicóptero. Desde el punto de vista de control ejemplifica un sistema de orden alto con acoplamientos cruzados muy significativos.

En el capítulo 10 presentamos la aplicación a la detección de marcas artificiales para la localización de robots, empleando para ello como sensores una cámara con zoom motorizado sobre una plataforma 'pan and tilt' y un anillo periférico de 24 sensores periféricos. Para fusionar toda la información y así poder corregir la posición y orientación del vehículo utilizamos un filtro de Kalman extendido basado en RGO.

Algunas partes de esta tesis han sido publicadas previamente:

La mayor parte (excepto la comparación con las redes neuronales) del capítulo 5 apareció como:

Garrido,S.,Moreno,L.E.,Salichs,M.A.(1998). Nonlinear On-line Identification of Dynamic Systems with Restricted Genetic Optimization. *6th European Congress on Intelligent Techniques & Soft Computing. (EUFIT'98)*

El capítulo 6 fue publicado como:

Garrido,S.,Moreno,L.E.,Balaguer,C.(1998). State Estimation for Non-linear Systems using Restricted Genetic Optimization. *11th Int. Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.(IEA-AIE'98)*

El capítulo 7 apareció como:

Garrido,S.,Moreno,L.E.,Salichs,M.A.(1999). Identification of State-Space Models with RGO. *6th European Congress on Intelligent Techniques & Soft Computing. (EUFIT'98)*

Las aplicaciones a los Robots Móviles fueron desarrolladas junto con otros autores en:

Armingol, J.M., Moreno, L.E., Garrido, S., de la Escalera, A., Salichs, M.A.: Mobile Robot Localization Using a Non-Linear Evolutionary Filter. *3rd IFAC Symp. on Intelligent Autonomous Vehicles (IAV'98)*.

1. FUNDAMENTOS Y ESTADO DEL ARTE.

1.1 Identificación

La identificación de sistemas es la teoría y el arte de construir modelos matemáticos de sistemas dinámicos basándonos en las entradas y salidas observadas. Como disciplina científica data de los primeros intentos de modelar series de tiempo usando técnicas AR. Aunque una parte sustancial del desarrollo de las técnicas está ligado a la Comunidad de Control, está básicamente construida a partir de técnicas estadísticas, en particular en los métodos de regresión lineal y no-lineal.

Construir modelos para sistemas desconocidos es un objetivo importante de la Ingeniería de control. Estos modelos necesitan simular el comportamiento real en los casos en que existe un conocimiento previo limitado de la estructura del sistema.

Consideremos el motor de un coche por ejemplo. Es importante simular el comportamiento de un motor para la detección de fallos y para propósitos de diagnóstico. Algunos factores importantes que afectan este proceso son las dificultades asociadas con el desarrollo de un modelo aceptable con un orden de complejidad mínimo y con un número de medidas mínimo. Estos factores hacen que el proceso de modelado de un motor de coche bastante difícil. Esto puede ser generalizado a una amplia clase de sistemas en la industria.

La identificación de sistemas no lineales se considera un problema difícil. La razón es que identificar un sistema no-lineal conlleva dos grandes etapas: la selección de la estructura del modelo con un cierto número de parámetros y la selección de un algoritmo que estime dichos parámetros.

En la literatura han sido propuestos numerosos modelos lineales como solución al proceso de identificación. Estos modelos son simples, como OE y ARX por ejemplo. En este caso podemos elegir un algoritmo sencillo para estimar los parámetros del modelo.

Aunque muchos sistemas no-lineales pueden ser linealizados representándolos por medio de una ecuación diferencial o en diferencias, modelar sistemas no-

lineales con modelos lineales implica muchas aproximaciones. Con frecuencia, estas aproximaciones no reflejan suficientemente el comportamiento real del sistema no lineal. Por tanto, el obtener un buen modelo, con una estructura que refleje la información real del sistema, exige un incremento en coste. Este coste es debido a la necesidad de algoritmos más avanzados que puedan manejar modelos con estructuras complejas.

Los modelos pueden ser paramétricos, que tienen la ventaja de estar dados por un conjunto pequeño de coeficientes, o bien no paramétricos como las redes neuronales, que tienen la ventaja de no estar restringidas a un cierto número, posiblemente pequeño, de descripciones posibles del modelo.

El término *Identificación de sistemas* fue acuñado por Lofti Zadeh [109] en 1962, como:

Identificación es la determinación, en base a la entrada y la salida, de un sistema, dentro de una clase de sistemas especificada, al cual el sistema probado es equivalente.

Puede decirse que la identificación de sistemas quedó establecida como un campo de investigación reconocido dentro del área de control automático a mediados de los sesenta: en el tercer congreso de la IFAC en Londres, 1966 en el que fue presentado un artículo de visión general sobre identificación de sistemas (Eykhoff *et al.* 1966). Un año después fue organizado el primer Symposium IFAC sobre identificación de sistemas en Praga. En la actualidad es el Symposium con una serie mas larga de la IFAC.

La teoría sobre identificación de sistemas está reunida, por ejemplo, en los libros de Ljung [69] y de Söderstrom y Stoica [97], donde también se pueden encontrar numerosas referencias.

1.1.1 Modelos de Sistemas Dinámicos

Cuando trabajamos con un modelo de un sistema, tenemos una cierta idea de cómo unas variables están relacionadas entre sí. Desde un punto de vista general, podemos decir que las relaciones observadas son un modelo del sistema. Está claro que los modelos pueden venir dados por diferentes estructuras y con distintas formulaciones matemáticas. El uso deseado del modelo también determina el grado de sofisticación requerido. El modelado e identificación de sistemas no-lineales es, en buena medida, un problema dependiente de las aplicaciones y que frecuentemente tiene sus raíces en la tradición y en las técnicas específicas del área de aplicación.

Un problema típico Observamos las entradas $u(t)$ y las salidas $y(t)$ de un sistema, con $t = 1, \dots, N$. Deseamos construir el modelo de un sistema, y podemos buscar un modelo sencillo de la forma

$$y(t) + a_1 y(t-1) = b_1 u(t-1) + b_2 u(t-2) \quad (1.1)$$

Es necesario determinar los valores de los parámetros a_1 , b_1 , y b_2 . Esto puede ser hecho por medio del conocido método de mínimos cuadrados

$$\min_{a_1, b_1, b_2} \sum_{t=1}^N (y(t) + a_1 y(t-1) - b_1 u(t-1) - b_2 u(t-2))^2 \quad (1.2)$$

Los valores \hat{a}_1^N , \hat{b}_1^N , y \hat{b}_2^N pueden ser calculados fácilmente en este caso puesto 1.2 es una función cuadrática. Así obtenemos el modelo del sistema:

$$y(t) + \hat{a}_1^N y(t-1) = \hat{b}_1^N u(t-1) + \hat{b}_2^N u(t-2) \quad (1.3)$$

Este sencillo problema de identificación de sistemas es un caso especial de la amplia clase de problemas de construcción de modelos.

1.1.2 Modelos de Entrada-Salida

Un modelo de entrada-salida describe un sistema dinámico basándose en los datos de entrada y de salida. Este tipo de modelos supone que la salida del sistema puede ser predicha a partir de las entradas y salidas pasadas del sistema. Si el sistema se supone, además determinista, invariante en el tiempo, de una entrada - una salida (SISO), el modelo de entrada-salida es:

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)) \quad (1.4)$$

donde $u(k)$, $y(k)$ representa el par de entrada-salida en el tiempo k . Los enteros positivos n y m son el número de salidas pasadas (también llamado el orden del sistema) y el número de entradas pasadas. En la práctica m es, normalmente, menor o igual que n . f puede ser cualquier función no-lineal definida desde el espacio de entradas y salidas pasadas hasta el espacio de salidas futuras.

Si el sistema es lineal, f es una función lineal y la ecuación anterior puede ser reescrita como:

$$\begin{cases} y(k) = a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n) \\ \quad + b_1 u(k-1) + b_2 u(k-2) + \dots + b_m u(k-m) \end{cases} \quad (1.5)$$

1.1.3 Modelos Paramétricos Lineales

El estado actual de la estimación de parámetros en la identificación de sistemas corresponde a la teoría clásica de la regresión. Es una necesidad para el futuro de la identificación de sistemas comprender en profundidad los relevantes desarrollos estadísticos en análisis de la regresión, como los expuestos en los libros de Carroll y Ruppert [15] y de McCullagh y Nelder [79].

Algunos de los modelos lineales mas conocidos son:

- ARX (Auto-Regressive with eXogenous inputs), cuya expresión es

$$A(q^{-1})y(t) = B(q^{-1})u(t - nk) + e_k \quad (1.6)$$

, que puede ser visto como una manera de determinar el siguiente valor de la salida dadas las observaciones anteriores y las entradas. Aquí $A(q^{-1})$ y $B(q^{-1})$ son polinomios en el operador desplazamiento hacia atrás q^{-1} e $y(t)$, $u(t)$, y $e(t)$ son las salidas, entradas y ruido, respectivamente. El ruido $e(t)$ es una sucesión aleatoria normalmente distribuida de media cero y varianza σ^2 .

- OE (Output Error) cuya expresión es

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t - nk) + e(t) \quad (1.7)$$

con la fuente de error $e(t)$ que es la diferencia (error) entre la salida real y la salida libre de ruido (teórica).

- ARMAX (Auto-Regressive Moving Average with eXogenous inputs), cuya expresión es

$$A(q^{-1})y(t) = B(q^{-1})u(t - nk) + C(q^{-1})e(t) \quad (1.8)$$

En éste modelo la expresión $A(q^{-1})y(t) = e(t)$ representa la auto-regresión, $y(t) = C(q^{-1})e(t)$ representa la media móvil de ruido blanco, mientras que $B(q^{-1})u(t)$ representa la entrada externa. Otros modelos pueden ser usados, como por ejemplo la representación en el espacio de estados o el modelo de ceros y polos.

1.1.4 Modelos Paramétricos No-lineales

La identificación de modelos matemáticos en forma de modelos lineales de una entrada - una salida está bien desarrollada en cuanto a la determinación de parámetros y determinación de la estructura. Sin embargo, se considera difícil aplicar con precisión la identificación de sistemas lineales a sistemas no-lineales. Un punto de vista pesimista es mirar a los sistemas no-lineales como el complementario de la clase de sistemas para los cuales existen métodos sistemáticos de identificación.

Algunos resultados interesantes son: el artículo de Johnson [56] sobre regresión no-paramétrica y el libro de Grenander [39]

Supongamos que se ha escogido una base del espacio de funciones $\{g_k\}$, debemos intentar aproximar la verdadera relación por una combinación lineal finita de las funciones de la base:

$$\hat{y}(t|\theta) = g(\phi(t), \theta) = \sum_{k=1}^n \theta(k) g_k(\phi(t)) \quad (1.9)$$

donde $\phi(t)$ está formado por las entradas $u(k)$ y salidas $y(k)$ pasadas:

$$\phi(t) = [y(t-1), \dots, y(t-n_a), u(t-1), \dots, u(t-n_b)]^T \quad (1.10)$$

y hemos introducido la notación $\hat{y}(t|\theta)$ para hacer hincapié en que $g(\phi(t), \theta)$ es una estimación de $y(t)$ dada la información $\phi(t)$ y un vector de parámetros dado θ .

El mejor valor de θ está determinado entonces por el conjunto de datos

$$Z^N = \{[y(t), \phi(t)] | t = 1, \dots, N\} \quad (1.11)$$

y será:

$$\hat{\theta}_N = \underset{\theta}{\operatorname{argmin}} \sum_{k=s}^N |y(t) - \hat{y}(t|\theta)|^2 \quad (1.12)$$

Modelos paramétricos No-lineales

Serie de Volterra Las series de Volterra son desarrollos en serie que consisten en términos lineales, bilineales y trilineales. Términos de mayor orden pueden ser incluidos para algunas aplicaciones. Una serie de Volterra con un orden y un retardo apropiadamente escogidos pueden ser usados para modelar sistemas

dinámicos no-lineales. Las series de Volterra vienen descritas por la expresión:

$$\begin{aligned}
 y(k) = & \sum_{i=0}^{\infty} a_i u(k-i) \\
 & + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} b_{ij} u(k-i) u(k-j) \\
 & + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{l=0}^{\infty} c_{ijl} u(k-i) u(k-j) u(k-l) \\
 & + \dots
 \end{aligned} \tag{1.13}$$

El vector u representa la señal de entrada, el vector y la salida del modelo. Para sistemas estables las series infinitas pueden ser truncadas sin problemas si el límite superior de la suma es lo suficientemente alto. Entonces la ecuación anterior puede ser simplificada como:

$$\begin{aligned}
 y(k) = & \sum_{i=0}^{N-1} a_i u(k-i) \\
 & + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{ij} u(k-i) u(k-j) \\
 & + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} c_{ijl} u(k-i) u(k-j) u(k-l) \\
 & + \dots
 \end{aligned} \tag{1.14}$$

De hecho, la mayoría de los estudios en éste área se limitan a series de Volterra de segundo o tercer orden.

Las series de Volterra pueden ser representadas en notación vectorial:

$$y(k) = a_1^T x_1 + b_2^T x_2 + c_3^T x_3 = x^T a \tag{1.15}$$

donde el vector de datos x y el vector de coeficientes viene dado por:

$$\begin{aligned}
 x^T = & (x(k), x(k-1), \dots, x(k-N+1), \\
 & x^2(k), x(k)x(k-1), \dots, x^2(k-N+1), \\
 & x^3(k), \dots, x^3(k-N+1))
 \end{aligned} \tag{1.16}$$

y

$$\begin{aligned}
 a^T = & (\underbrace{a_0, \dots, a_{N-1}}_{a_1^T}, \underbrace{b_{00}, \dots, b_{N-1, N-1}}_{b_2^T}, \\
 & \underbrace{c_{000}, \dots, c_{N-1, N-1, N-1}}_{c_3^T})
 \end{aligned} \tag{1.17}$$

Los parámetros desconocidos de la serie pueden ser determinados a partir de las medidas de entrada-salida resolviendo la ecuación

$$y = Xa \tag{1.18}$$

donde

$$y = \begin{pmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-N) \end{pmatrix}$$

$$X = \begin{pmatrix} x_1^T(k) & x_2^T(k) & x_3^T(k) \\ x_1^T(k-1) & x_2^T(k-1) & x_3^T(k-1) \\ \vdots & \vdots & \vdots \\ x_1^T(k-N) & x_2^T(k-N) & x_3^T(k-N) \end{pmatrix}$$

$$a = \begin{pmatrix} a_1 \\ b_2 \\ c_3 \end{pmatrix}$$

La solución por mínimos cuadrados viene dada por la ecuación normal $X^T X a = X^T y$ que tiene como solución:

$$\hat{a} = (X^T X)^{-1} X^T y \quad (1.19)$$

Modelos NARMAX Entre los modelos no lineales, uno de los mas interesantes, es el modelo NARMAX (Non- linear Auto-Regressive Moving Average with eXogenous inputs), que es una generalización de las series de Volterra y que puede ser descrito como

$$y(t) = F[y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), \epsilon(t-1), \dots, \epsilon(t-n_\epsilon)] + \epsilon(t) \quad (1.20)$$

donde t representa el tiempo discreto, $y(t)$, $u(t)$ y $\epsilon(t)$ son la salida, la entrada, y el error de predicción, respectivamente, n_y , n_u y n_ϵ son los órdenes correspondientes, $F[\cdot]$ es una función no lineal y d es el mínimo retardo de la entrada [Leontaritis 85]. El modelo polinomial NARMAX tiene como expresión

$$y(k) = a_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_y} \sum_{j=1}^{n_u} c_{ij} y(k-i) u(k-j) \quad (1.21)$$

nos permite una descripción buena y sencilla de una amplia clase de sistemas no lineales.

Por ejemplo, si el modelo exacto NARMAX es

$$y(k) = y(k-1) e^{-u(k-1)} \quad (1.22)$$

podemos desarrollar en serie la exponencial

$$\exp(-x) = \sum_{i=0}^{\infty} \frac{(-x)^i}{i!} \quad (1.23)$$

y emplear un modelo polinomial aproximado NARMAX

$$y(k) = y(k-1) - y(k-1)u(k-1) + \frac{1}{2}y(k-1)u^2(k-1) - \frac{1}{6}y(k-1)u^3(k-1) \quad (1.24)$$

1.1.5 Modelos No-Paramétricos, Redes Neuronales

Los modelos de Redes Neuronales artificiales han sido estudiados durante muchos años con el ánimo de conseguir prestaciones parecidas a las humanas en campos como el reconocimiento del habla o de imagen. Estos modelos están compuestos por muchos elementos computacionales trabajando en paralelo y organizados en patrones que recuerdan a las redes neuronales biológicas.

Los elementos computacionales o nodos están conectados mediante pesos que son adaptados a lo largo del proceso para mejorar sus prestaciones. Es imposible cubrir todos los tipos de redes neuronales y solamente vamos a considerar las llamadas Feedforward y Recurrentes, que son las más utilizadas en identificación de sistemas.

Redes Feedforward

El paso desde el desarrollo general 1.9 de una función en la base $\{g_k\}$ a lo que es una red neuronal no es grande. Basta escoger $g_k(\phi) = \alpha_k \sigma(\beta_k \phi + \gamma_k)$ donde β_k es un vector de parámetros y γ_k y α_k son parámetros escalares para obtener

$$g(\phi) = \sum_{k=1}^n \alpha_k \sigma(\beta_k \phi + \gamma_k) + \alpha_0 \quad (1.25)$$

donde α_0 es el valor correspondiente al nivel medio. A este modelo se le denomina Red Feedforward con una capa oculta y una salida.

Las funciones de la base, llamadas nodos o neuronas, son funciones univalueadas que convierten la red neuronal en un desarrollo en funciones simples. La elección específica de $\sigma(\cdot)$ es denominada función de activación y suele ser escogida igual para todos los nodos.

El nombre *feedforward* está explicado por la figura; hay una dirección específica en el flujo de los cálculos cuando se calcula la salida g . Primero se calculan

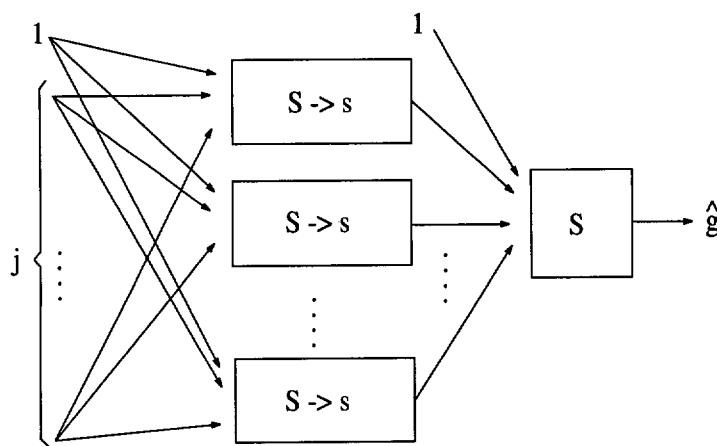


Fig. 1.1: Red Feedforward con una capa oculta y una salida.

las sumas con peso en la entrada de cada unidad, después estas sumas son pasadas por la función de activación y forman las salidas de las capas ocultas. Para calcular g se forma una suma con pesos de las salidas de los nodos de la capa oculta. Si g es una función vectorial, hay varios nodos de salida formando la capa de salida. La entrada ϕ se denomina, a veces, capa de entrada. Los pesos de las diferentes sumas son los parámetros de la red.

La función $\sigma(x)$ se toma de tipo *sigmoide*, es decir es una función continua que verifica $\lim_{x \rightarrow \infty} \sigma(x) = a$ y $\lim_{x \rightarrow -\infty} \sigma(x) = b$ con $a, b \in \mathbb{R}$ y $b < a$.

La red neuronal de una capa oculta 1.25 se puede generalizar a varias capas ocultas. La salida de la primera capa oculta alimenta la segunda capa y así sucesivamente, y la última capa oculta alimenta a la capa de salida, como se puede ver en la figura. La fórmula para dos capas ocultas será

$$g(\phi) = \sum_i \theta_{1,i}^3 \sigma \left(\sum_j \theta_{i,j}^2 \sigma \left(\sum_m \theta_{r,m}^1 \phi_m \right) \right) \quad (1.26)$$

Redes Recurrentes

Si alguna de las entradas de la red *feedforward* consiste en salidas retardadas de la red, o bien en algún estado interno retardado, la red recibe el nombre de red recurrente o red dinámica. Éste tipo de redes es especialmente interesante para identificación.

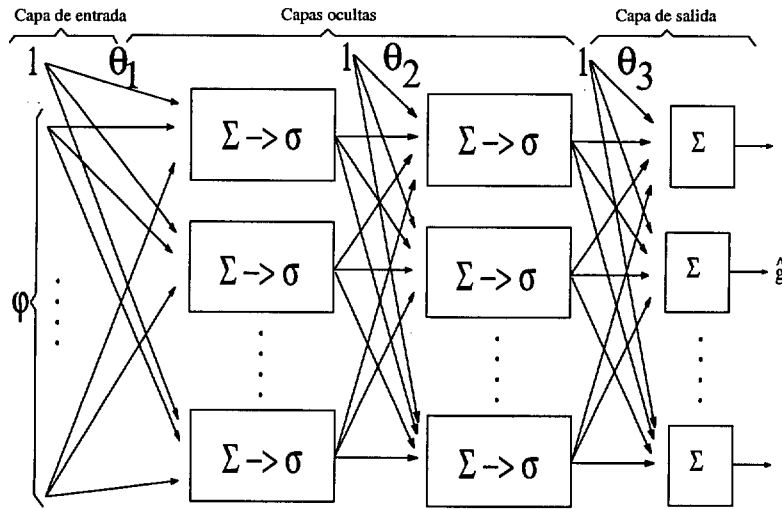


Fig. 1.2: Red Feedforward con dos capas ocultas.

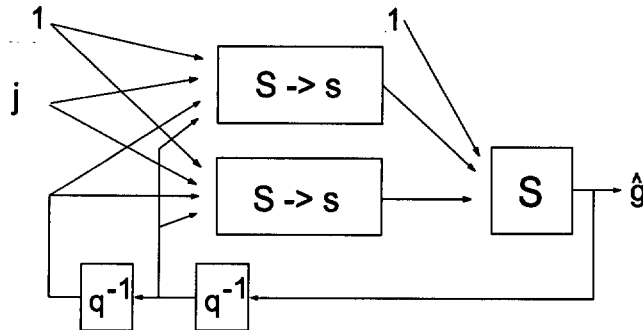


Fig. 1.3: Red Recurrente. q^{-1} retarda la señal un tiempo de muestreo.

Aspectos Algorítmicos

Los pesos de la red deben ser escogidos de forma que la salida se ajuste lo mejor posible a los datos observados, para ello hay que minimizar el funcional:

$$V_N(\theta) = \frac{1}{2N} \sum_{t=1}^N |y(t) - g(\phi(t), \theta)|^2 \quad (1.27)$$

Como no hay solución analítica a este problema, es necesario emplear algún método numérico de optimización. Las rutinas de búsqueda más eficientes están

basadas en búsqueda local en una dirección de mayor pendiente desde el punto actual. Tenemos, por tanto, un esquema iterativo del tipo

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} - \mu_i R_i^{-1} \nabla \hat{g}_i \quad (1.28)$$

donde μ_i es el paso, $\nabla \hat{g}_i$ es una estimación del gradiente $V'_N(\hat{\theta}^{(i)})$ y R_i es una matriz que modifica la dirección de búsqueda.

Identificación con Redes Neuronales

Los dos esquemas tradicionales de identificación para sistemas dinámicos, llamados serie-paralelo y paralelo, son los representados en las figuras 1.4 y 1.5, respectivamente.

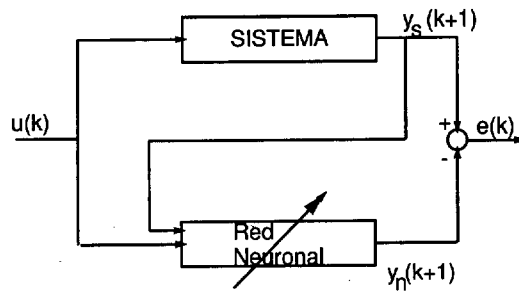


Fig. 1.4: Identificación Serie-Paralelo

El esquema de identificación serie-paralelo utiliza las entradas y salidas de la planta y deja que internamente la red deduzca el error existente. Este esquema, también llamado *teacher forcing* puede ser usado con cualquier algoritmo de aprendizaje.

La idea es que durante la fase de entrenamiento, para calcular los nuevos estados de la red neuronal, se utilicen en el lazo de realimentación las salidas actuales de la planta en lugar de las salidas de la red. Su implementación puede ser muy sencilla con cualquier algoritmo.

El serie-paralelo es útil al inicio del aprendizaje para que los valores de la red no se alejen mucho y converja más rápido.

En el esquema de identificación paralela, la red no tiene información directa acerca de las salidas de la planta, sólo las entradas y el error entre las salidas de la planta y la red.

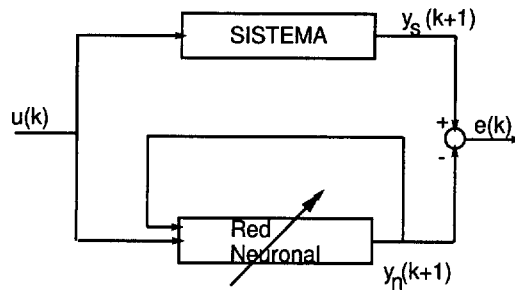


Fig. 1.5: Identificación Paralela

Redes neuronales con paralelización del tiempo La representación del tiempo y su influencia en el comportamiento de los sistemas es uno de los problemas mas difíciles en las aplicaciones de modelado y control.

Una forma de representar eventos temporales es su representación directa como una dimensión explícita en un vector del espacio de entrada. El primer elemento representa el primer evento temporal, el segundo elemento de cada vector representa el segundo evento temporal y así sucesivamente.

La representación del tiempo en las redes neuronales utilizadas en aplicaciones de sistemas dinámicos se pueden clasificar en dos grupos de métodos distintos.

El primer grupo tiende a diseñar su propia arquitectura de red dinámica en aplicaciones específicas. El segundo grupo lo forman las modificaciones a las estructuras y a los algoritmos de aprendizaje estático, para incluir en ellos de manera explícita el nuevo parámetro tiempo y así poder representar la dinámica del sistema.

Los retardos pueden ser introducidos en la capa de entrada o en las capas internas.

Ejemplos de estas arquitecturas son:

Time Delay Neural Networks (TDNN):

Hopfield y Tank [45] (1986) , Toma de decisiones y Optimización.

Guyon y Albrecht [40] (1990), Reconocimiento de caracteres escritos a mano.

Redes Neuronales Recurrentes :

Rumelhart, Hinton y Williams [94] (1986), Completar secuencias.

Williams y Zipser (1989), Reconocimiento de caracteres.

Robinson y Fallside (1988), Reconocimiento de voz.

Memory Neuron Networks :

Poddar [89] (1991): Reconocimiento de caracteres.

El segundo grupo está formado por las modificaciones a las estructuras y a los algoritmos de aprendizaje estáticos para incluir en ellos el tiempo.

Una de las más conocidas es la red de la figura 1.6 que fue propuesta por Elman (1990). En ella se aprecia que la capa de entrada está dividida en dos partes ; el conjunto de entradas externas y una copia de la activación de la capa oculta en el instante anterior.

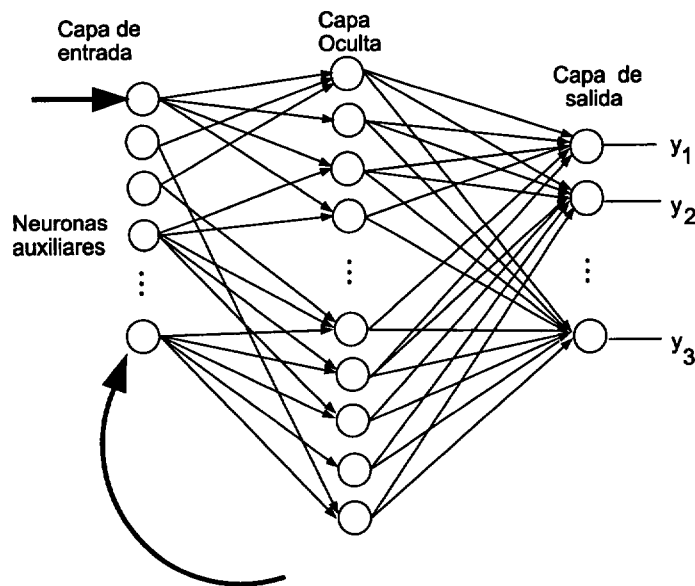


Fig. 1.6: Red con realimentación de la Capa Oculta.

La red es de propagación hacia adelante y al ajustar los pesos hay que tener en cuenta los pesos que van desde las entradas auxiliares a la capa oculta. Este tipo de red puede reconocer o reproducir secuencias de periodos cortos de tiempo. Este tipo de red se utilizó para el reconocimiento de palabras generadas por una gramática formal.

Jordan propuso una arquitectura 1.7 donde la capa de entrada está dividida en dos partes: el conjunto de entradas externas y la realimentación de la activación de la capa de salida a través de conexiones de valor fijo.

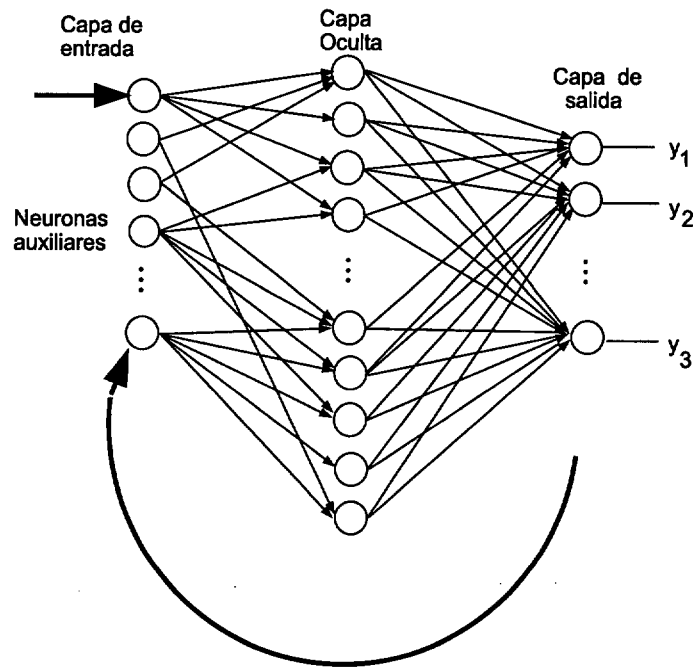


Fig. 1.7: Red con realimentación de la Capa de Salida.

Este tipo de red es utilizada para el reconocimiento y clasificación de secuencias. Fue sugerida por Jordan para su empleo en la generación de trayectorias de robots.

Ejemplos de otras arquitecturas dentro de éste grupo son:

- *Backpropagation in Dynamic Systems (BPD):*

[Narendra [84] (1991)]: Identificación y Control, Control por Modelo de Referencia.

Propone la construcción de una segunda red llamada 'sensitivity model' para el cálculo en línea de las derivadas de las salidas con respecto a los parámetros (matriz Jacobiana) y de las derivadas con respecto a las entradas. En el artículo, por lo demás explica con detalle el esquema propuesto para generar el gradiente.

- *Backpropagation through time (BTT):*

[Werbos [104] (1990)]: Identificación, Camión remolque.

El fundamento consiste en permitir a la red observar lo que ocurrirá con el sistema durante un tiempo T de pasos futuros, para que pueda adaptar sus pesos en consecuencia. Debido a la duplicación de unidades, los pesos y el coste computacional del entrenamiento, hacen que esta arquitectura no sea muy popular.

1.1.6 Identificación con Algoritmos Genéticos

La mayoría de los métodos usuales en la identificación de sistemas están basados en mínimos cuadrados o en el método de máxima verosimilitud que son métodos locales guiados por el gradiente y necesitan una función de pérdidas (loss function) diferenciable y un espacio de búsqueda suave.

Esto tiene el inconveniente de que si las señales tienen ruido coloreado, como suele ser habitual en la práctica, los métodos basados en mínimos cuadrados tienen el fracaso asegurado.

Si además estamos implementado el método en línea el ruido estará correlacionado con los regresores y el método explotará, fenómeno conocido como blow-up.

Para evitar esto en algunos casos sencillos, se puede usar un factor de olvido muy próximo a la unidad, pero en la mayoría de los casos de sistemas no lineales variantes en el tiempo esta técnica falla.

A menudo fallan en la búsqueda de óptimos globales cuando éste espacio de búsqueda es no diferenciable, o no lineal en sus parámetros. Los métodos de optimización basados en algoritmos genéticos pueden mejorar los resultados porque no utilizan las propiedades de diferenciabilidad o de ser lineal en los parámetros.

En 1992, Kristinsson y Dumont [64] hicieron un primer estudio de la identificación de sistemas y del control off-line con algoritmos genéticos de sistemas dinámicos continuos y discretos. Comprobaron que los algoritmos genéticos son efectivos en ambos dominios y que es posible identificar directamente parámetros físicos, o bien, polos y ceros. Hicieron simulaciones con sistemas de fase mínima y no mínima y de un sistema con dinámica no modelada.

En 1993, Flockton y White [25] utilizan los algoritmos genéticos para hacer otra identificación off-line de polos y ceros con algunas pequeñas mejoras.

Posteriormente, en 1995, Tan, Li y Murray-Smith [99] usan los algoritmos genéticos con "simulated annealing" para hacer una identificación off-line y la linealización de sistemas no lineales, invariantes del tiempo con un modelo ARMAX. Utiliza esta técnica de búsqueda para identificar los parámetros de un sistema descrito por un modelo ARMAX en variables de estado, en presencia de ruido blanco y para aproximar un sistema no lineal multivariable por un modelo en el espacio

de estados, lineal e invariante en el tiempo.

En este mismo año, Iba, de Garis y Sato [50] usan una técnica numérica off-line, que integra una búsqueda adaptativa de una estructura de árbol basada en programación genética y un mecanismo de ajuste local de parámetros empleando búsqueda estadística. En la programación genética tradicional, la recombinación causa frecuentes rupturas de los bloques de construcción y la mutación puede causar abruptos cambios en la semántica. Para superar estas dificultades ellos implementan la programación genética tradicional con una búsqueda local "hill-climbing", usando un procedimiento de ajuste de parámetros.

Todos estos métodos trabajan off-line y precisan que el sistema sea invariante del tiempo. El último además utiliza un método no paramétrico.

1.1.7 Identificación en Bucle Cerrado

La *Identificación para el Control* ha tenido notable interés en los últimos años (ver, por ejemplo, los artículos de Gevers [34], Lee [67], Van den Hof [23] y la tesis de Forssell [27]). El objetivo es encontrar modelos válidos para un control adaptativo. El uso de experimentos en bucle cerrado es una de las características de estos métodos. Otra de las razones para su utilización puede ser la inestabilidad de la planta.

La finalidad de la identificación en Bucle Cerrado es obtener buenos modelos del sistema en bucle abierto, a pesar de la realimentación.

Varios métodos que dan estimaciones consistentes para datos en bucle abierto pueden fallar cuando se aplican de manera directa a la identificación en bucle cerrado. Estos incluyen los análisis espectral y de correlación, el método de la variable instrumental, los métodos de subespacios y los OE (output error) con modelo de ruido incorrecto (ver la tesis de Forssell [27] y el artículo de Forssell y Ljung [28]).

Consideramos el sistema

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (1.29)$$

con $e(t)$ ruido blanco de varianza λ_0 y el regulador es

$$u(t) = r(t) - F_y(q)y(t). \quad (1.30)$$

El sistema en bucle cerrado puede ser escrito como:

$$y(t) = G_o(q)S_o(q)r(t) + S_o(q)H_o(q)e(t) \quad (1.31)$$

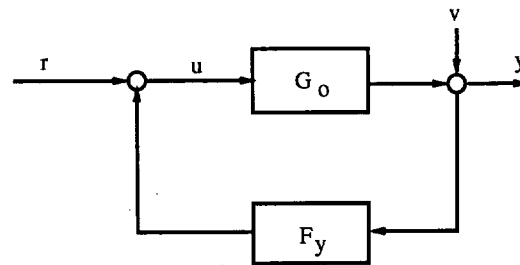


Fig. 1.8: Sistema en bucle cerrado.

donde $S_o(q)$ es la función de sensibilidad,

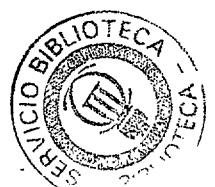
$$S_o(q) = \frac{1}{1 + F_y(q)G_o(q)} \quad (1.32)$$

Es importante saber que el método de error de predicción puede ser aplicado directamente, como si la realimentación no existiera, y el resultado dará una precisión óptima si el sistema verdadero puede ser descrito dentro de la estructura de modelo escogida. Sin embargo, debido a las trampas a que nos puede llevar la identificación en bucle cerrado, han sido sugeridos diversos métodos alternativos. Se pueden clasificar en (ver el artículo de Gustafsson [41]):

- *Procedimiento Directo:* Se aplica el método de error de predicción y se identifica el sistema en bucle abierto usando medidas de la entrada y la salida.
- *Procedimiento Indirecto:* Se identifica el sistema en bucle cerrado utilizando medidas de la señal de referencia r y de la salida y y se usa esta estimación para hallar los parámetros del sistema en bucle abierto utilizando el conocimiento del controlador.
- *Procedimiento de Entrada-Salida conjuntas:* Se identifican las funciones de transferencia de r a y y de r a u y se utilizan para hallar una estimación del sistema en bucle abierto.

$$\begin{aligned} y(t) &= G_o(q)S_o(q)r(t) + S_o(q)H_o(q)e(t) \\ u(t) &= S_o(r(t) - F_y(q)S_o(q)H_o(q)e(t)) \end{aligned} \quad (1.33)$$

El procedimiento de Identificación Directa es el método mas natural para hacer identificación en bucle cerrado. Las principales razones son:



- Funciona bien a pesar de la complejidad del regulador y no requiere conocimiento de la realimentación.
- No se necesitan algoritmos ni software especiales. Se asegura la consistencia y la precisión óptima siempre que la estructura del modelo contenga al sistema verdadero.

Los problemas que plantea éste procedimiento son:

- Es necesario tener buenos modelos del ruido (ver teorema 8.4 en el libro de Ljung [69]).
- Un pre-filtro/modelo de ruido que cambie las características del ruido verdadero produce sesgo.

El problema básico de los datos en bucle cerrado es que contienen menos información que cuando se toman los datos en bucle abierto, puesto que un importante propósito de la realimentación es hacer al sistema en bucle cerrado insensible a los cambios del sistema en bucle abierto.

Los métodos de error de predicción, aplicados de una manera directa, con modelo de ruido que puede describir las propiedades del ruido verdadero dan estimaciones consistentes y precisión óptima. Además no se necesita conocimiento de la realimentación. Por esto deben ser vistos como los métodos de primera elección.

Este procedimiento de Identificación directa es el utilizado en esta tesis, teniendo en cuenta las precauciones anteriores.

1.2 Estimación de Estados

Cuando el estado de un sistema tiene que ser estimado a partir de información sensorial con ruido, se necesita emplear algún tipo de estimador de estados para fusionar los datos de los diferentes sensores y producir una estimación precisa del verdadero estado del sistema.

Si la dinámica del sistema y el modelo de observación son lineales, entonces el estimador de mínimos cuadrados puede ser calculado usando el *filtro de Kalman*.

1.2.1 Modelos en el Espacio de Estados

Un sistema dinámico puede ser también descrito por un modelo en el espacio de estados. El modelo en el espacio de estados de orden n de múltiples entradas y

múltiples salidas (MIMO), no-lineal e invariante en el tiempo tiene la forma:

$$\begin{cases} x(k+1) = \phi(x(k), u(k)) \\ y(k) = \psi(x(k)) \end{cases} \quad (1.34)$$

donde:

$$\begin{cases} x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T \\ u(k) = (u_1(k), u_2(k), \dots, u_m(k))^T \\ y(k) = (y_1(k), y_2(k), \dots, y_l(k))^T \end{cases} \quad (1.35)$$

$x(k)$, $u(k)$, e $y(k)$ son el vector de estados, el vector de entradas y el vector de salidas, respectivamente. Si el sistema es lineal,

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (1.36)$$

donde A , B , C y D son matrices $(n \times n)$, $(m \times n)$, $(n \times l)$ y $(m \times l)$ respectivamente.

Estas estructuras sirven para modelar tanto sistemas lineales como no-lineales.

1.2.2 La estimación lineal en sistemas dinámicos - El Filtro de Kalman

El problema de estimación dinámica

Consideremos un sistema dinámico lineal en tiempo discreto descrito por una ecuación vectorial en diferencias con ruido blanco Gaussiano aditivo que modela “perturbaciones no predecibles”. La ecuación de estado es

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k), \quad k = 0, 1, \dots \quad (1.37)$$

donde $x(k)$ es el vector de estado de dimensión n_x , $u(k)$ es el vector de entrada de dimensión n_u (que es conocido), y $v(k)$, $k = 0, 1, \dots$, es la secuencia de ruido blanco Gaussiano de media cero con covarianza

$$E[v(k)v(k)'] = Q(k) \quad (1.38)$$

La ecuación de medida es

$$z(k) = H(k)x(k) + w(k), \quad k = 1, 2, \dots \quad (1.39)$$

con $w(k)$ secuencia de ruido de medida blanco Gaussiano de media cero con covarianza

$$E[w(k)w(k)'] = R(k) \quad (1.40)$$

Las matrices F , G , H , Q , y R se suponen conocidas y que pueden cambiar con el tiempo. En otras palabras, el sistema puede cambiar con el tiempo y los ruidos son no estacionarios.

El estado inicial $x(0)$ es en general desconocido y se modela como una variable aleatoria Gaussiana con media y covarianza conocidas. Las dos secuencias de ruido y el estado inicial se suponen mutuamente independientes.

Las anteriores condiciones se llaman lineales-Gaussianas.

El algoritmo de estimación El algoritmo de estimación comienza con una estimación inicial $\hat{x}(0|0)$ de $x(0)$ y la covarianza inicial asociada $P(0|0)$, que se supone disponible.

Un ciclo del algoritmo de estimación dinámica (filtro de Kalman) consiste en estimar a partir

$$\hat{x}(k|k) = E[x(k)|Z^k] \quad (1.41)$$

que es la media condicional del estado en el tiempo k . dadas las observaciones hasta el tiempo k inclusive, y de la matriz de covarianza asociada

$$P(k|k) = E[(x(k) - \hat{x}(k|k))[x(k) - \hat{x}(k|k)]' | Z^k] \quad (1.42)$$

las correspondientes variables del siguiente paso, $\hat{x}(k+1|k+1)$ y $P(k+1|k+1)$.

Esto se puede hacer puesto que las variables Gaussianas quedan caracterizadas por sus dos primeros momentos.

1.2.3 El caso no lineal: Filtros No-lineales

En el caso no lineal el problema de filtrado en el espacio de estados viene dado por

$$x(k) = g(k, x(k-1), \varepsilon(k)), \quad (1.43)$$

$$z(k) = h(k, x(k), \eta(k)), \quad (1.44)$$

donde $\varepsilon(k)$ y $\eta(k)$ son los ruidos, que se suponen vectores aleatorios, independientemente distribuidos:

$$\begin{pmatrix} \varepsilon(k) \\ \eta(k) \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix} \right) \quad (1.45)$$

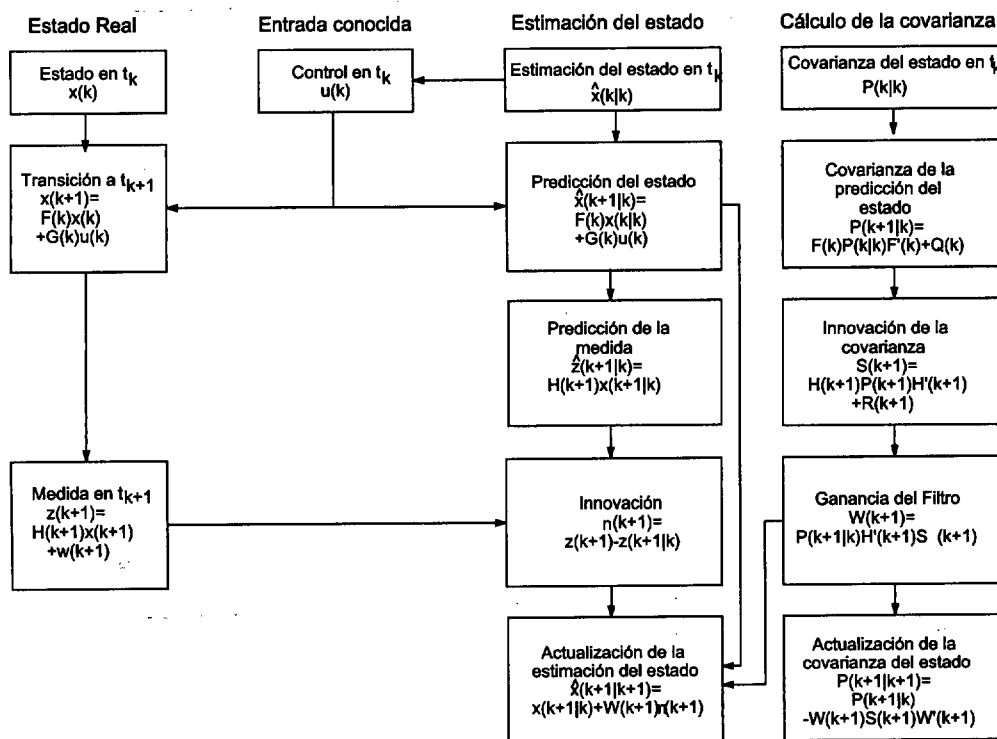


Fig. 1.9: Diagrama de flujo del filtro de Kalman.

La estimación óptima implica la descripción de la densidad de probabilidad condicional

$$\hat{x}^{MMSE}(k|k) = E(x(k)|Z^k) = \int x(k)p(x(k)|Z^k)dx . \quad (1.46)$$

Desafortunadamente, esta descripción necesita un número infinito de parámetros. Por esta razón, un número importante de aproximaciones subóptimas han sido propuestas. Estos métodos utilizan aproximaciones analíticas de las distribuciones de probabilidad, la ecuación de transición de estados o la ecuación de medida. Existen otros métodos, tales como el método de Montecarlo, que necesitan miles de puntos para aproximar la densidad de probabilidad condicional. En aplicaciones con muchas dimensiones, éstos últimos métodos no son prácticos. Por esta razón son necesarios métodos con un número razonable de operaciones, tales como los estudiados en este trabajo.

1.2.4 Filtro Extendido de Kalman (EKF) y Filtro No-lineal de Segundo Orden (SNF)

El filtro Extendido de Kalman es similar a un filtro de Kalman linealizado, con la excepción de que la linealización es realizada en la trayectoria estimada en vez de hacerlo en una trayectoria nominal previamente calculada. Por esta razón, las funciones $g(k, x(k-1), \varepsilon(k))$ y $h(k, x(k), \eta(k))$ son desarrolladas en serie de Taylor alrededor de $\hat{x}(k|k)$ con términos hasta primer o segundo orden para obtener el EKF o SNF respectivamente. El desarrollo con términos de segundo orden de la ecuación de transición es:

$$\begin{aligned}
 x(k) &= g(k, x(k-1), \varepsilon(k)) \simeq g(k, \hat{x}(k|k-1), 0) \\
 &+ g_x(k, x(k-1), \varepsilon(k))(x(k-1) - \hat{x}(k-1|k-1)) + g_\varepsilon(k, x(k-1), \varepsilon(k))\varepsilon(k) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' g_{xx}^j(k, x(k-1), \varepsilon(k)) (x(k-1) - \hat{x}(k-1|k-1)) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j \varepsilon'(k) g_{\varepsilon\varepsilon}^j(k, x(k-1), \varepsilon(k)) \varepsilon(k) \\
 &+ \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' g_{x\varepsilon}^j(k, x(k-1), \varepsilon(k)) \varepsilon(k) .
 \end{aligned} \tag{1.47}$$

y el desarrollo con términos hasta segundo orden de la ecuación de medida es:

$$\begin{aligned}
 z(k) &= h(k, x(k), \eta(k)) \simeq h(k, \hat{x}(k|k-1), 0) \\
 &+ h_x(k, x(k-1), \eta(k))(x(k-1) - \hat{x}(k-1|k-1)) + h_\eta(k, x(k-1), \eta(k))\eta(k) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' h_{xx}^j(k, x(k-1), \eta(k)) (x(k-1) - \hat{x}(k-1|k-1)) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j \eta'(k) h_{\eta\eta}^j(k, x(k-1), \eta(k)) \eta(k) \\
 &+ \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' h_{x\eta}^j(k, x(k-1), \eta(k)) \eta(k) .
 \end{aligned} \tag{1.48}$$

donde e_j es el j^{th} vector de la base Cartesiana.

1.2.5 Filtro extendido Iterado de Kalman (IEKF)

La predicción de la medida, hasta primer orden, es $\hat{z}(k|k-1) = h(k, \hat{x}(k|k-1))$. Hay errores en predicción en usar $\hat{x}(k|k-1)$ para $x(k)$. Otros errores adicionales

son debidos a la no linealidad en la medida. Es posible disminuir estos errores si el estado actualizado no es calculado como una aproximación a la esperanza condicional sino como un estimador del tipo maximum a posteriori.

La función de densidad de probabilidad condicional, PDF, de $x(k)$ dado Z^k , si todas las variables aleatorias son Gaussianas, es

$$\begin{aligned} p(x(k)|Z^k) &= p(x(k)|z(k), Z^{k-1}) \\ &= \frac{1}{c} p(z(k)|x(k)) p(x(k)|Z^{k-1}) \\ &= \frac{1}{c} \mathcal{N}(z(k); h(k, x(k)), R(k)) \mathcal{N}(x(k); \hat{x}(k|k-1), P(k|k-1)) . \end{aligned} \quad (1.49)$$

Maximizar esta función es equivalente a minimizar la siguiente

$$\begin{aligned} V(x(k)) &= \frac{1}{2} (z(k) - h(k, x(k)))' R(k)^{-1} (z(k) - h(k, x(k))) \\ &\quad + \frac{1}{2} (x(k) - \hat{x}(k|k-1))' (P(k|k-1))^{-1} (x(k) - \hat{x}(k|k-1)) . \end{aligned} \quad (1.50)$$

El Filtro de Kalman Extendido Iterado (IEKF) usa un algoritmo Newton-Raphson para estimar $\hat{x}(k|k)$. Desarrollando V en serie de Taylor hasta segundo orden alrededor de la i -ésima estimación de $x(k)$ da como resultado:

$$V = V^i + V_x^i (x - x^i) + \frac{1}{2} (x - x^i)' V_{xx}^i (x - x^i) . \quad (1.51)$$

Poniendo el gradiente a cero:

$$x^{i+1} = x^i - (V_{xx}^i)^{-1} V_x^i . \quad (1.52)$$

Esto da lugar a la ecuación IEKF:

$$\begin{aligned} \hat{x}^i(k|k) &= \hat{x}^{i-1}(k|k) + P^{i-1}(k|k) H^{i-1}(k)' R(k)^{-1} \{z(k) - h(k, \hat{x}^{i-1}(k|k))\} \\ &\quad - P^{i-1}(k|k) P(k|k-1)^{-1} (\hat{x}^i(k|k) - \hat{x}(k|k-1)), \end{aligned} \quad (1.53)$$

with $H^{i-1}(k) = h_x(k, \hat{x}^i(k|k))$.

1.3 Validación de Modelos

Después de la estimación, la pregunta obvia es si el modelo que se ha derivado es adecuado para el uso que se le pretende dar. Éste es el difícil y subjetivo problema de la validación. De acuerdo con Bohlin la mejor manera de atacar el

problema es confrontar al modelo con todas las fuentes de información disponibles, incluyendo el conocimiento previo, los datos experimentales y la experiencia usando el modelo. Para tener más confianza en el modelo el consejo general es emplear tantas herramientas de validación como sea posible.

La primera forma concebible es el uso del sentido común. Si, por ejemplo, un parámetro estimado representa la longitud de un eje, ésta debe ser positiva. Otro ejemplo puede ser el nivel de un depósito. Los tests de éste tipo pertenecen a la categoría del conocimiento previo y son especialmente importantes cuando los parámetros tienen algún significado físico.

La gran mayoría de los métodos están basados en datos experimentales. Un test básico consiste en calcular la varianza de los parámetros estimados. Una varianza grande comparada con el valor del parámetro indica que algo está mal. Otra posibilidad es estimar varios modelos en paralelo. La respuesta en frecuencia del modelo paramétrico puede ser comparada con la respuesta espectral, por ejemplo.

La herramienta de validación más versátil, en todas las categorías, es la simulación. El sistema verdadero y el sistema derivado se alimentan con la misma señal de entrada y se comparan las salidas. Para una buena comparación es deseable que el experimento se base en datos nuevos, es decir que no se hayan utilizado para la estimación. Esto recibe el nombre de validación cruzada.

También suele ser valioso investigar la secuencia de residuos $\{\varepsilon(\hat{\theta}_N)\}$, especialmente con datos nuevos. Una manera muy sencilla es representar gráficamente estos residuos y comprobar si la secuencia resultante muestra el patrón de ruidos esperado. Conjuntamente con esto, para saber si los residuos son más o menos blancos, puede ser estudiada su covarianza muestral

$$R_N^{\varepsilon}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t, \hat{\theta}_N) \varepsilon(t + \tau \hat{\theta}_N),$$

que para $\tau > 0$ debe ser pequeño, al menos si N es grande. Además la independencia entre la entrada y la salida puede ser comprobada representando

$$R_N^{\varepsilon, u}(\tau) = \frac{1}{N} \sum_{t=\max(1, 1-\tau)}^{\min(N, N-\tau)} \varepsilon(t, \hat{\theta}_N) u(t + \tau),$$

para varios retardos τ . Éstos deben ser también pequeños, puesto que de lo contrario, hay más información en la salida del sistema real, originada por la entrada. Lo que significa que todavía existen dinámicas no modeladas.

1.4 Control Adaptativo

Los problemas de control en los sistemas no-lineales han sido extensivamente estudiados en el pasado. No obstante, existe un gran interés en la comprensión de la conducta de los sistemas no-lineales en cuanto a la modelización e identificación de la estructura y de los parámetros de tales sistemas cuando se diseña un controlador adaptativo.

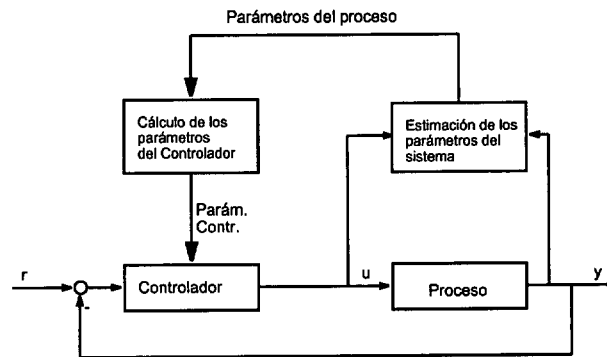


Fig. 1.10: Sistema adaptativo con identificación de modelo (MIAS)

El Control adaptativo trata el problema de controlar la salida de la planta en presencia de incertidumbres paramétricas o estructurales. En el control adaptativo tradicional, para conseguir que un problema fuera tratable, la planta se suponía lineal con parámetros desconocidos. Se escogía una estructura de controlador, y los parámetros se ajustaban escogiendo una ley adaptativa, de modo que la salida de la planta siguiera a la referencia asintóticamente.

Desde el principio de la tecnología de control adaptativo, se han propuesto dos clases distintas de controladores adaptativos. En el indirecto, los parámetros de la planta se estiman y se ajustan en base a los datos de entrada-salida. En el directo, los parámetros del controlador se ajustan directamente en base a los datos de entrada -salida.

Existen una gran variedad de esquemas adaptativos dentro de estas dos clases, tales como el Model Reference Adaptive Control (MRAC), Self-Tuning Adaptive Control (STAC), Self-Organizing Fuzzy Logic (SOFLIC), Neural Networks (NN), y Neurofuzzy Adaptive Control (NAC).

Para los procesos cuyos parámetros varían lentamente en el tiempo, los controladores adaptativos con realimentación pueden ser divididos en dos grandes grupos. Los sistemas adaptativos con identificación de modelo (MIAS) determi-

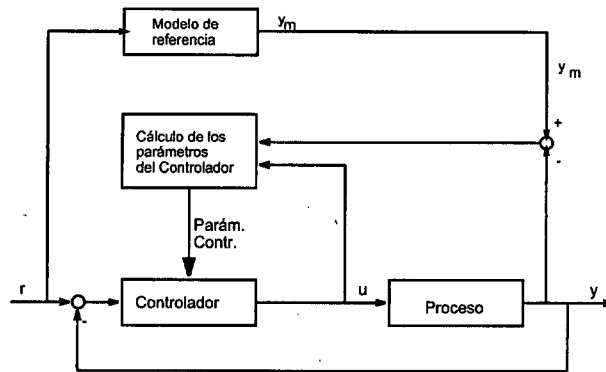


Fig. 1.11: Sistema adaptativo con modelo de referencia (MRAS)

nan un modelo del proceso las medidas de entrada-salida y métodos de identificación. Aquí los parámetros son calculados de acuerdo con el método de diseño del controlador que ha sido programado con anterioridad. Esta clase de reguladores adaptativos también es denominada 'self tuning regulators'.

Los sistemas adaptativos con modelo de referencia (MRAS) intentan obtener una respuesta en bucle cerrado próxima a la dada por el modelo de referencia para la señal de entrada. Aquí una señal externa, por ejemplo la variable de referencia, es medida y la diferencia entre las señales se forma usando las señales del bucle de control y del modelo de referencia y cambiando los parámetros del controlador por medio de un método adaptativo.

A lo largo de esta tesis hemos utilizado controladores adaptativos con identificación de modelo(MIAS).

2. CONCEPTOS BÁSICOS DE LOS A.G.

2.1 Algoritmos Genéticos

Introducción

Los algoritmos genéticos son mecanismos de búsqueda basados en las leyes de la selección natural y de la genética. Combinan la supervivencia de los individuos mejor adaptados junto con un intercambio aleatorio de información, lo que le da al algoritmo un aire de búsqueda biológica.

Los algoritmos genéticos han sido desarrollados por John Holland y su equipo de estudiantes de la universidad de Michigan. Las metas que se proponían eran: explicar rigurosamente los procesos adaptativos y diseñar nuevos sistemas artificiales que sigan manteniendo los mecanismos de los sistemas naturales. Para más detalles se pueden consultar [Goldberg 91] y [Davis 89].

Se parte, inicialmente, de un conjunto de individuos, la población, y de una función que evalúa su eficacia para el problema concreto y que es la función que se desea optimizar. Al valor que se asigna a cada individuo a partir de esta función se le denomina salud.

- John Holland, (1975). "Adaptation in natural and artificial systems."
- Algoritmos que manejan poblaciones consistentes en soluciones codificadas de problemas.
- La búsqueda de buenas soluciones es realizada en el espacio de soluciones codificadas.
- Manipulación de poblaciones: selección, cruce y mutación.

Fig. 2.1: Algoritmos Genéticos.

En cada generación un conjunto de nuevos individuos se forma utilizando par-

tes de la información de los cromosomas de los individuos mejor adaptados de la generación anterior. Los algoritmos genéticos no son sólo una simple búsqueda aleatoria, ya que utilizan de forma eficiente la información histórica de las generaciones anteriores para indagar nuevos caminos de búsqueda que mejoren los resultados anteriores.

Las características principales son:

- Los algoritmos genéticos no trabajan directamente con los objetos, sino con la codificación de esos objetos, que pueden ser un número, un conjunto de parámetros, etc. A esta codificación le llamamos cromosoma. Concretamente, en nuestro trabajo, hemos representado a cada individuo por un cromosoma definido por una sucesión de 0 y 1, que representan los coeficientes del modelo paramétrico.
- Los algoritmos genéticos realizan la búsqueda mediante toda una generación objetos, no buscan un único elemento.
- Los algoritmos genéticos utilizan una función objetivo que nos da la información de lo adaptados que están los individuos, y no las derivadas u otra información auxiliar.

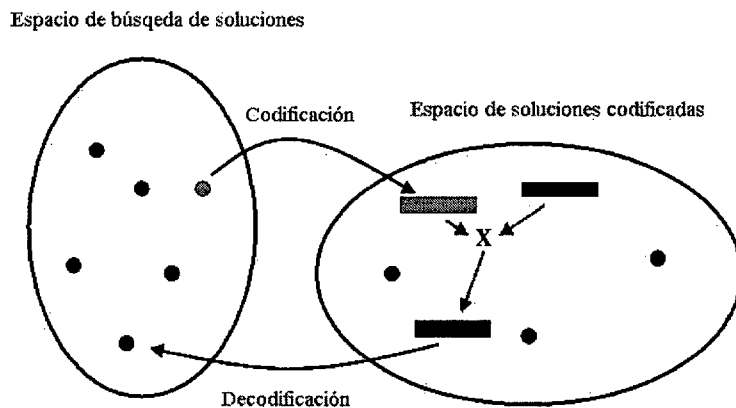


Fig. 2.2: Búsqueda Genética.

- Las reglas de transición son probabilísticas, no determinísticas.
- Los mecanismos de los algoritmos genéticos simples son sorprendentemente fáciles, dado que los mecanismos que utiliza son el mecanismo de copia y el de intercambio de partes de las cadenas de cromosomas.

El por qué este mecanismo funciona es algo más sutil, y el efecto combinado de simplicidad de la operación y la potencia de su efecto son dos de las principales atracciones de la aproximación mediante algoritmos genéticos.

Algunos de los principales atractivos de los algoritmos genéticos como método de optimización son:

- Se pueden utilizar como método de optimización global o semi-local, según interese.

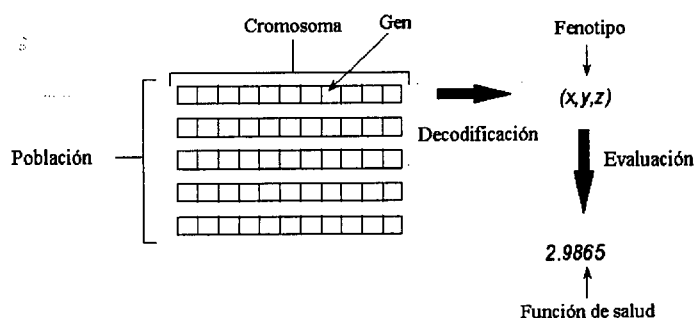


Fig. 2.3: Terminología de los Algoritmos Genéticos.

- No hace uso de ningún gradiente, ni conceptos similares. Por lo que se puede usar con señales con mucho ruido, en las que no tiene sentido las ideas asociadas a la derivada. No obstante, si las funciones son derivables, los algoritmos genéticos se pueden usar en la primera fase de minimización, para encontrar las cuencas de atracción, y a continuación utilizar, por ejemplo, un método basado en el gradiente.
- Se pueden utilizar en línea.

Los principales inconvenientes son:

- El número de operaciones y por tanto el coste computacional puede ser muy alto si no se toman precauciones.
- No hay ninguna teoría matemática hasta el momento, que nos asegure la convergencia.
- Se puede emplear fácilmente computación en paralelo.



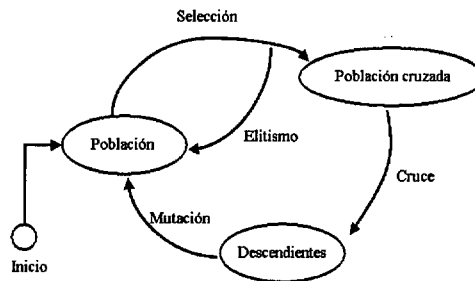


Fig. 2.4: Ciclo de los Algoritmos Genéticos.

Como es lógico, este tipo de métodos son apropiados cuando trabajamos con funciones que no son suaves y no son convexas y que no se deben aplicar a casos sencillos en que son más rápidos y precisos otros métodos, como los basados en el gradiente.

- ¿Son los AG un intercambio aleatorio de bits? ¿Qué hay detrás?
- Holland creó un teorema llamado ‘teorema de los esquemas de Holland’.
- Existen otros teoremas, algunos de ellos basados en el análisis de las cadenas de Markov:
 - ¿Es una cadena de diferentes soluciones que permite encontrar la solución óptima?

Fig. 2.5: ¿Cómo funcionan los Algoritmos Genéticos?

2.1.1 Algoritmo Genético Simple

En un algoritmo genético la información ha de codificarse para poder trabajar adecuadamente con ella. Existen numerosos sistemas de codificación, aunque nosotros utilizaremos uno de los más sencillos: el código binario natural. Este código a pesar de su simplicidad cumple la teoría de los esquemas de Holland en cuanto a tener el menor número posible de caracteres (o alelos): únicamente dos. Una vez definido el sistema de codificación a emplear vamos a ver como actúan los operadores básicos de selección, cruce y mutación sobre este código.

Selección: Supongamos una población de puntos codificados en cadenas binarias (limitamos el tamaño de la población a cuatro muestras):

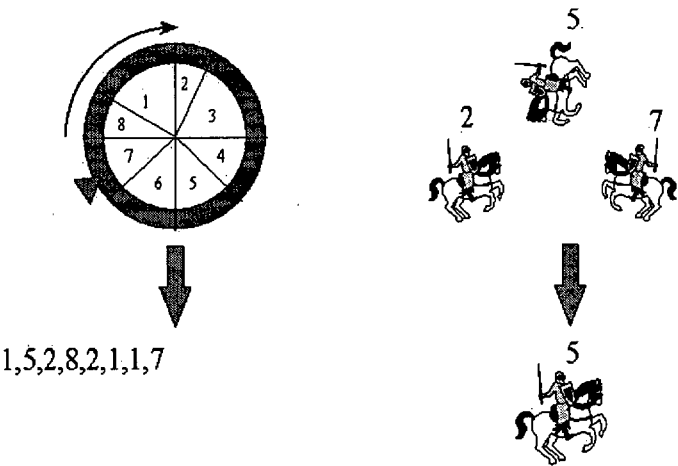


Fig. 2.6: Métodos de selección.

Individuo Codificado	Valor(Fenotipo)	Salud asociada
01101	13	169
10100	20	400
00100	4	16
00111	7	49

Tab. 2.1: Fenotipos y salud asociada.

El valor decimal (fenotipo) de cada cadena codificada así como la salud que nos devuelve la función problema $F(x)=x^2$ están reproducidas en la tabla anterior. Vamos a definir la probabilidad de selección de cada individuo como el cociente entre su salud y la suma de la salud de todos los individuos (salud global, de manera que aquellos individuos con mayor salud tengan un mayor porcentaje de selección. El número de veces esperado que cada cadena sea seleccionada para reproducirse no es más que su probabilidad por el tamaño de la población (así el tamaño de la población generada será igualmente de cuatro muestras).

Individuo	Probabilidad	Nº Copias Esperado
01101	$169/634= 0,27$	1,1
10100	$400/634= 0,63$	2,5
00100	$16/634= 0,02$	0,1
00111	$49/634= 0,08$	0,3

Tab. 2.2: Número de copias esperado.

El mecanismo de selección, de manera gráfica, es como sigue: supongamos una ruleta dividida en cuatro porciones proporcionales a la probabilidad de cada individuo, donde al individuo de mayor salud (mayor probabilidad) se le asigna la mayor porción. Si elegimos un punto de la ruleta al azar, éste estará comprendido dentro de una de las porciones (cuanto más grande sea la porción más probable es que el punto elegido al azar caiga dentro de ella), seleccionando para la reproducción al individuo asociado a la misma. Así, los individuos de mayor salud (mayor porción) tienen una mayor probabilidad de ser seleccionados, mientras los de poca salud (poca porción de ruleta) corren una suerte contraria.

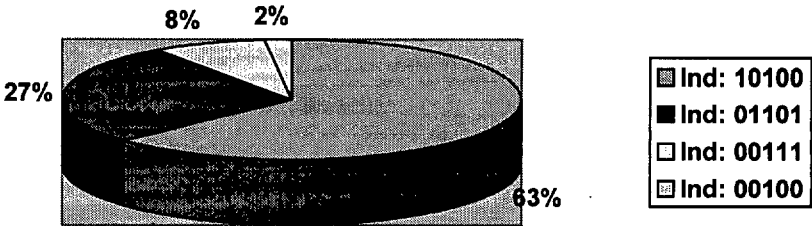


Fig. 2.7: Función de salud.

El número de veces que debemos repetir el proceso de la ruleta es igual al tamaño de la población que deseamos generar, dado que por cada pareja de padres seleccionados obtendremos en la reproducción una pareja de hijos.

Cruce Una vez seleccionados dos padres, el cruce ocurre con una probabilidad P_c (entre 0 y 1, normalmente 0,4 a 0,6) fijada de antemano en el algoritmo. Si se cumple P_c hemos de elegir al azar un punto de cruce en las cadenas binarias que representan a los padres, de manera que obtengamos por combinación dos individuos hijos con la misma longitud de cadena que los padres. En el caso de no cumplirse la probabilidad de cruce tendríamos dos hijos exactos a los padres.

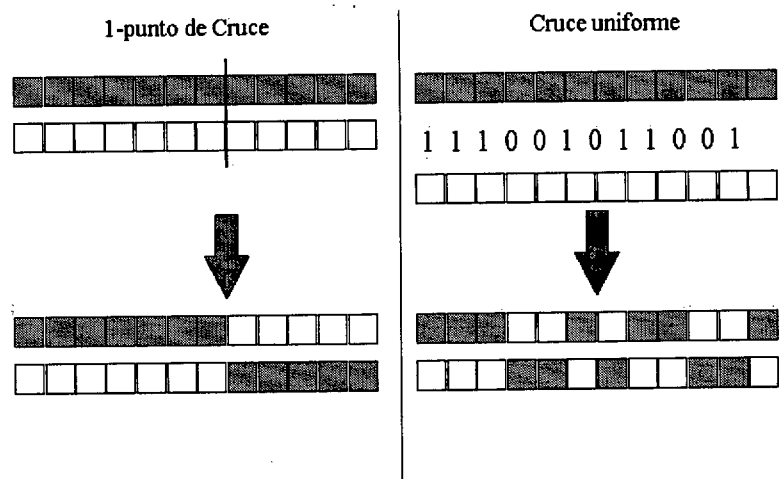


Fig. 2.8: Cruce.

Por ejemplo, si en la selección tenemos como padres a los individuos 10100 y 00111, con una probabilidad de cruce $P_c=0,6$ y en una tirada al azar entre 0 y 1 obtenemos un 0,3 ponemos en marcha el mecanismo de cruce, debiendo elegir al azar un punto de cruce en las cadenas (en nuestro caso éste debe limitarse entre las posiciones 1 y 4, donde existe cadena binaria). Si conseguimos, por ejemplo, un 3 al azar y contamos de derecha a izquierda, los padres quedarían divididos en dos subcadenas por el tercer bit:

10100 →	10 100	PADRE 1
00111 →	00 111	PADRE 2

Tab. 2.3: Cruce

Los hijos resultarán de una combinación de estas cuatro subcadenas: el primer hijo contendrá los tres primeros bits del Padre 1 y los dos últimos del Padre 2, mientras el segundo hijo llevará la información relativa a los tres primeros bits del Padre 2 y los dos últimos del Padre 1.

Hay que notar que en este mecanismo no se consideran como posibles puntos de cruce el bit 0 y el bit 5 (extremos de las cadenas), ya que tendríamos

	PADRE1	PADRE 2	
Hijo 1 →	.. 100	+ 00 ...	→ 00100
Hijo 2 →	10 ...	+ .. 111	→ 10111

Tab. 2.4: Cruce

como hijos copias idénticas de los padres, lo que equivaldría a no haber cruce.

Mutación: El operador de mutación, encargado de mutar alelos para conseguir una mayor riqueza genética y no perder información, actúa en nuestro algoritmo genético básico bajo una probabilidad de mutación P_m fijada de antemano (de valor muy bajo). Cada vez que generemos nuevas poblaciones, tras el mecanismo de cruce debemos correr el de mutación para todos los alelos de cada individuo. Si se cumple P_m el alelo cambia de valor, de 0 a 1 y viceversa.

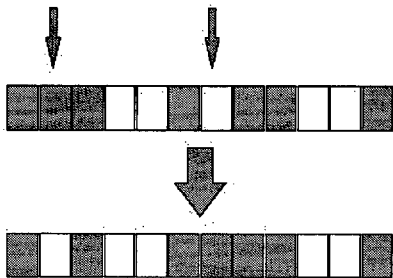


Fig. 2.9: Mutación.

Supongamos uno de los individuos generados en el ejemplo del mecanismo de cruce básico: 00100, mediante tiradas al azar chequeamos una posible mutación en cada alelo con una P_m dada (por ejemplo de 0,01). Si se produce mutación en el alelo 1 y en el 3, la cadena binaria quedaría modificada como sigue: 00001, cambiando así el fenotipo del individuo (originalmente correspondía al valor 4; tras la mutación ha pasado al valor 1).

Esto habría que repetirlo con toda la población generada.

2.1.2 Mecanismos Avanzados

Reescalado: Un problema frecuente en los algoritmos genéticos es el de la convergencia prematura, donde se tiende a alcanzar un valor muy cercano al óptimo buscado en las primeras generaciones, para mantener después este valor a lo largo de la simulación sin llegar al mejor resultado posible. Este fenómeno es debido a una alta competitividad inicial entre individuos, permitiendo a aquellos con una salud excepcional dominar sobre el resto, y a un decremento de ésta en generaciones posteriores, al irse hacinando la población en intervalos cada vez más reducidos.

Contra esta convergencia prematura existen mecanismos de reescalado consistentes en modificar la salud de los individuos para optimizar en cada momento su competitividad, a fin de lograr una solución de equilibrio entre competitividad y eficiencia.

Reescalado lineal: es el más utilizado y simple. La salud escalada viene definida por:

$$f' = af + b \quad (2.1)$$

con 'f' la antigua función de salud. Siendo ' C_{mult} ' el número esperado de copias del mejor individuo de cada población, las constantes 'a' y 'b' se eligen según:

$$f'_{max} = C_{mult} \cdot f_{media} \quad (2.2)$$

$$f'_{media} = f_{media} \quad (2.3)$$

De esta manera reducimos el intervalo de salud en las primeras generaciones (disminuyendo su alta competitividad) y lo aumentamos en las últimas (elevando la baja competitividad), logrando una mayor eficiencia y retardando la convergencia.

El problema del reescalado lineal aparece en aquellas situaciones en que es necesario disminuir la salud de individuos con una salud de por sí baja respecto a la media. En este caso puede ocurrir que los nuevos valores escalados sean negativos. Para prevenir esto limitamos la salud escalada mínima a cero.

Reescalado por desviación típica: sugerido por Forrest en 1985, utiliza la desviación típica de la población para definir la función de reescalado:

$$f' = f - (f - c \cdot \sigma) \quad (2.4)$$

donde la constante 'c' toma un valor múltiplo de la desviación típica σ , normalmente entre 1 y 3.

Reescalado por potencias: Gilles, en 1985, experimentó con otro método de reescalado donde la salud escalada quedaba como potencia de la salud original:

La variable k depende del problema en particular, siendo necesario modificarla a lo largo de la simulación.

Otros Métodos de Selección

Además del mecanismo básico de selección, implementado en el algoritmo genético simple, existen otras versiones más complejas y especializadas. Entre ellas cabe destacar:

Muestreo estocástico sin reemplazo: es el método de selección experimentado por De Jong en su algoritmo de prueba R3, basado en valores esperados de selección.

Muestreo determinístico: cada individuo es seleccionado un número de veces igual a la parte entera de su valor de copias esperado ($p_{seleccion}n$), completando la población con aquellos individuos cuya parte fraccionaria de este valor sea mayor.

Muestreo estocástico permanente con/sin reemplazo: tienen gran parecido al método anterior (determinístico) aunque difieren en el modo de completar la población de selección. En el muestreo estocástico permanente con reemplazo la parte fraccionaria del valor esperado de copias sirve para calcular el peso que cada individuo tendrá en una selección básica por ruleta. Por otra parte, en el muestreo estocástico permanente sin reemplazo, estas partes fraccionarias se utilizan directamente como probabilidades de selección a la hora de completar la población.

Muestreo estocástico por competición: sugerido a Brindle por Wetzel en 1983.

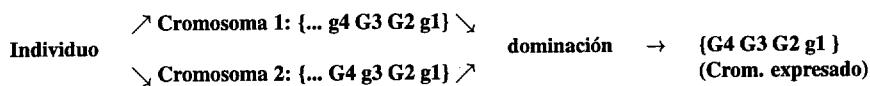
Inicialmente se realiza una selección de pares de individuos por el mecanismo básico, calculando las probabilidades de selección de la manera habitual, para después introducir en la nueva población el individuo con mayor salud de cada pareja, desechando el otro.

Aunque Brindle en sus estudios no pudo concluir cuál de los anteriores era el mejor método (todos tenían parecida eficiencia, aunque siempre superior al mecanismo básico de selección), Booker (1982) demostró en investigaciones sobre aprendizaje automático la superioridad del muestreo estocástico permanente sin reemplazo.

Representación Dual y Dominancia

Si observamos la naturaleza gran parte de las especies asocian su genotipo a un par de cromosomas, donde ciertos alelos (dominantes) dominan sobre otros (recesivos), de manera que el fenotipo se determina por combinación de estos dos cromosomas y por predominio de alelos. Esta representación dual permite a los individuos tener memoria de información genética pasada, y que puede ser de utilidad en cualquier otro momento. La dominancia se encarga de seleccionar de entre esta información la que se precisa en el momento actual. Así, si se produjera un cambio brusco en el medio ambiente de una especie, ciertas combinaciones genéticas que la sirvieron en el pasado podrían ser de nuevo las que determinasen su supervivencia.

En el siguiente ejemplo hemos representado los dos cromosomas que definen al individuo como una cadena de genes que pueden tomar valores dominantes ('G' mayúscula) o recesivos ('g' minúscula). El número que los acompaña marca su posición dentro del cromosoma.



El cromosoma expresado es aquel que resulta de combinar el cromosoma 1 y el 2 aplicando el mecanismo de dominación: 'G4' del cromosoma 2 domina frente a 'g4' del cromosoma 1, resultando que en la posición 4 del cromosoma expresado tendremos 'G4'. En el caso de los genes de la posición 1 los dos son recesivos por lo que no dominaría ninguno de ellos, resultando que el primer gen del cromosoma expresado sería el 'g1'.

La representación dual es de gran efectividad en entornos cambiantes con el tiempo, donde aumenta la eficiencia y adaptación del algoritmo, mientras en entornos invariables no se perciben cambios significativos con el uso de este mecanismo genético.

En la historia de los algoritmos genéticos, estos operadores fueron utilizados por Bagley, Hollstien, Holland y Brindle.

Bagley incluyó en cada gen de la pareja de cromosomas un indicador de dominación junto a la información codificada del fenotipo del individuo. Hollstien (1971) desarrolló un sistema de dominación trialélico para códigos binarios introduciendo tres alelos, el 1, equivalente a un 1 dominante, el -1, equivalente a un 1 recesivo y el 0 que también es recesivo. El mapa de dominación es el siguiente:

	0	-1	1
0	0	0	1
-1	0	1	1
1	1	1	1

Tab. 2.5: Mapa de dominación.

Brindle experimentó con otros modelos de dominación:

Dominación variable y global: la probabilidad de dominación de un 1 ó 0 se determina a partir de su proporción en la población actual. Con estas probabilidades y mediante pruebas de Bernouilli obtenemos los alelos resultantes para cada gen de los cromosomas.

Dominación determinística. variable y global: el alelo dominante es aquel con mayor proporción en la población actual.

Selección aleatoria de un cromosoma: se seleccionan como alelos dominantes en sus respectivas posiciones todos los de un cromosoma elegido al azar de entre la población.

Dominación del mejor cromosoma : de cada pareja de cromosomas de un individuo tomamos como alelos dominantes los de aquel cromosoma con mayor salud.

Dominación controlada por un tercer cromosoma: el mapa de dominación se incluye para cada individuo en un tercer cromosoma, de manera que se pueda adaptar en las sucesivas generaciones. Este modelo tiene el inconveniente de no tener en cuenta las relaciones que pudieran establecerse entre los genes funcionales (correspondientes al genotipo) y aquellos que determinan la dominación. En los sistemas biológicos las leyes que rigen qué alelos y en qué momento son dominantes/recesivos están íntimamente ligadas a la información genética del individuo, y por tanto resulta inadecuado separar los genes de dominación de los funcionales.

Inversión: El operador de inversión actúa permutando las posiciones de los genes dentro de las cadenas codificadas de información que representan a los individuos. Se trata de un mecanismo de reordenación que intenta encontrar las mejores combinaciones posicionales de alelos, evitando la pérdida de información.

Su implementación en los algoritmos genéticos consiste en invertir la posición de un par de alelos seleccionados a partir de puntos de cruce. Así, elegimos al azar dos puntos de inversión de la cadena binaria, que queda dividida en tres subcadenas, permutando las posiciones del primer y último alelo de la cadena central.

Cromosoma: $\begin{matrix} \text{puntos de cruce} \\ \{ \dots g6 | g5 g4 g3 | g2 g1 \} \\ \{ \dots 1 | 0 1 1 | 1 0 \} \end{matrix} \rightarrow \text{inversión} \rightarrow \begin{matrix} \{ \dots g6 | g3 g4 g5 | g2 g1 \} \\ \{ \dots 1 | 1 1 0 | 1 0 \} \end{matrix}$

Hay que notar que, aunque debido al código de representación utilizado en nuestro algoritmo genético la inversión modifica el fenotipo del individuo, en los sistemas naturales la información contenida en los genes es independiente de su posición en el cromosoma y la inversión no tiene efecto alguno sobre su salud.

Un problema asociado al mecanismo de inversión en códigos binarios es el de cruce entre cromosomas no homólogos. Si cruzamos dos cromosomas no homólogos a los que se haya aplicado la inversión, puede ocurrir que las cadenas resultantes no contengan el mapa genético completo, reduciendo así la efectividad del mecanismo de cruce.

Cromosoma 1 (invertido): $\{ \dots g6 g3 | g4 g5 g2 g1 \}$ \searrow \nearrow $\{ \dots g6 g3 g3 g4 g2 g1 \}$
 Cromosoma 2 (invertido): $\{ \dots g6 g5 | g3 g4 g2 g1 \}$ \nearrow \searrow $\{ \dots g6 g5 g4 g5 g2 g1 \}$
 Cruce

En este ejemplo, debido a la recombinación de los genes por la inversión, los cromosomas resultantes tras el cruce no contienen a todos los genes (del g1

al g7) sino que algunos se encuentran repetidos. Así, podemos haber perdido información dado que estos cromosomas no poseen el mapa genético completo.

Bagley (1967) propuso restringir los cruces a cromosomas homólogos para evitar este efecto indeseado. En 1972 Frantz estudió dos modelos de inversión en funciones con distintos grados de no linealidad:

Inversión lineal: inversión simple (anteriormente explicada).

Inversión lineal y extrema: asignó una probabilidad de 0,75 a la inversión lineal, si ésta no se cumplía podía suceder una inversión de los alelos extremos del cromosoma (primero y último) con probabilidad 0,125. Esta variante intentaba contrarrestar el hecho de que la inversión tiende a permutar alelos centrales frente a los situados cerca de los extremos.

Frantz utilizó estos modelos según dos modos de funcionamiento:

Inversión continua: el mecanismo de inversión se activa según una cierta probabilidad p_j y se aplica a cada nuevo individuo en cuanto es generado.

Inversión masiva: no se aplica la inversión hasta que se ha generado toda la población, para después utilizar los dos mismos puntos de cruce (inversión simple) en la mitad de ésta.

Para evitar los problemas de cruce entre cromosomas no homólogos, Frantz experimentó con cuatro sistemas:

Cruce restringido: limitamos el cruce, tal como Bagley sugirió, a cromosomas homólogos.

Cruce por viabilidad: no se restringe el cruce, aunque si las cadenas resultantes no incluyen todo el mapa genético se desechan.

Cruce por patrones: tras seleccionar dos cromosomas se aplica la inversión a uno de ellos (al azar), y según el patrón de éste se invierte el segundo. El cruce se realiza, posteriormente, de la manera habitual.

Cruce por patrón del mejor cromosoma: parecido a la variante anterior, pero el cromosoma que define el patrón de inversión es el de mayor salud.

2.2 Algoritmos Genéticos con parámetros continuos

Esta técnica ha tenido en los últimos años una gran difusión y resuelve el problema de la cuantización que se daba en los algoritmos tradicionales.

Con la codificación binaria en los algoritmos genéticos es que normalmente no se aprovecha toda la precisión del ordenador. Cada parámetro necesita muchos bits para representarlo completamente. ¿Que se puede hacer si se quiere usar toda la precisión posible? Una posible respuesta es representar los parámetros en coma flotante. Cuando la variable es continua, ésta es la manera más natural de representar los números. Además se tiene la ventaja de que se requiere un menor tamaño de memoria que para el almacenamiento en binario.

Uno puede, en principio, utilizar cualquier representación concebible para representar los parámetros. Cuando los parámetros están naturalmente cuantizados, los algoritmos genéticos binarios se ajustan bien al problema. Sin embargo, cuando los parámetros son continuos es mas lógico representarlos por números en coma flotante.

Además, como los algoritmos genéticos binarios tienen su representación limitada a la representación binaria de los parámetros, utilizando números reales se consigue la precisión de la máquina de manera sencilla.

Otra de las principales ventajas de los algoritmos genéticos con parámetros continuos es la rapidez, ya que los operadores no suelen trabajar a nivel de bit como en el caso binario, sino que se trabaja a nivel de todo el número en coma flotante:

Selección Se ordenan los cromosomas con arreglo a su salud y nos quedamos con los mejores miembros de la población.

Cruce En los métodos más simples se escogen uno o más puntos en el cromosoma para marcar los puntos de cruce. Entonces los parámetros entre estos puntos simplemente se intercambian entre los dos padres.

Mutación Con una cierta probabilidad, que suele estar comprendida entre un 1% y un 20% se seleccionan los cromosomas que van a ser mutados. A continuación se seleccionan aleatoriamente los parámetros del cromosoma que van a ser mutados. Finalmente cada parámetro a mutar es reemplazado por otro nuevo parámetro aleatorio o bien se le suma otro nuevo parámetro aleatorio.

3. ESTUDIO MATEMÁTICO DE LOS A.G.

Es natural modelar los Algoritmos Genéticos como procesos o cadenas de Markov en los que el estado de un Algoritmo Genético está dado por el contenido de la población actual.

Uno puede imaginar el espacio de estados de todas las posibles poblaciones y estudiar las características de las trayectorias de la población (las cadenas de Markov) que un Algoritmo Genético produce a partir de poblaciones generadas aleatoriamente.

3.1 Introducción a los procesos de Markov

Consideremos un proceso que puede ser descrito en cada paso por uno de los N estados distintos i_1, i_2, \dots, i_N . En cada paso el sistema puede cambiar de estado (o permanecer en el mismo) de acuerdo con un conjunto de probabilidades de transición. Cuando estas probabilidades sólo dependen del estado de partida y de llegada y no de los estados anteriores, se dice que el proceso es de Markov.

Definición 3.1.1: Un proceso estocástico $\{X_k\}$, $k=1,2,\dots$ con un espacio de estados $S = \{i_1, i_2, i_3, \dots, i_N\}$ es un proceso (cadena) de Markov si para cada n y todos los estados i_1, i_2, \dots, i_N es cierto que

$$p(X_n = i_n | X_{n-1} = i_{n-1}, \dots, X_1 = i_1) = p(X_n = i_n | X_{n-1} = i_{n-1}) \quad (3.1)$$

Esta propiedad es equivalente a la ecuación

$$p(X_0 = i_0, \dots, X_n = i_n) = p(X_0 = i_0)p(X_1 = i_1 | X_0 = i_0) \dots p(X_n = i_n | X_{n-1} = i_{n-1}) \quad (3.2)$$

y también a la ecuación

$$p(AB | X_n = j) = p(A | X_n = j)p(B | X_n = j) \quad (3.3)$$

donde A es un suceso arbitrario definido por los resultados de los pasos $0, 1, \dots, n-1$ y B es un suceso definido por los resultados de los pasos $n+1, n+2, \dots$. Esta ecuación se describe a menudo como la *independencia entre el futuro y el pasado si el presente es conocido*.

En el caso de las cadenas de Markov, el sistema está descrito por una matriz P de dimensiones $N \times N$ que da la probabilidad de transitar del estado i en el tiempo $t-1$ al estado j en el tiempo t .

$$P(i, j)(t) = p_{i,j}(t) = P(X_t = j | X_{t-1} = i) \quad (3.4)$$

donde X_t es la variable aleatoria de la cadena de Markov.

Los valores $p_{i,j}(t)$ definen la matriz $P(i, j)(t)$ de probabilidad de transición de un estado a otro en un paso de tiempo.

Definición 3.1.2: La probabilidad de ir del estado i al estado j en el paso t se llama probabilidad de transición $p_{ij}(t)$. Si la probabilidad de transición es independiente de t , la cadena de Markov se llama homogénea.

Vamos a estudiar solamente cadenas de Markov homogéneas en un espacio $S = \{i_1, i_2, \dots, i_n\}$ finito, por tanto, las probabilidades de transición vienen dadas por una matriz de transición $P = (p_{ij})$ que verifica $p_{ij} \geq 0, \forall i = 1, 2, \dots, n, \forall j = 1, 2, \dots, n$. A las matrices con estas propiedades se las denomina estocásticas.

La matriz de transición en n pasos es la n -ésima potencia de P :

$$P_{i,j}^{(n)} = p_{i,j}^{(n)} = P(X_t = j | X_{t-n} = i) \quad (3.5)$$

Se verifica la ecuación

$$p_{i,i}^{(n+m)} = \sum_k p_{i,k}^{(n)} p_{k,j}^{(m)}, \forall n \geq 0, \forall m \geq 0 \quad (3.6)$$

que es llamada *ecuación de Markov*. También puede ser escrita en forma matricial como

$$P^{(m+n)} = P^{(m)} P^{(n)} \quad (3.7)$$

Es también posible calcular las probabilidades condicionales sobre un conjunto de estados J :

$$p_{i,J}^{(n)} = p(X_t \in J | X_{t-n} = i) = \sum_{j \in J} p_{i,j}^{(n)} \quad (3.8)$$

3.2 Clasificación de estados

Definición 3.2.1: Un estado j es *accesible* desde un estado i cuando se puede ir de i a j en un número finito de pasos.

Desde el punto de vista de la matriz de transición, esta definición implica

- $i \rightarrow j$ si para algún n , $P_{ij}^{(n)} > 0$.
- $i \leftrightarrow j$ si para algún n y algún m , $P_{ij}^{(n)} > 0$ y $P_{ji}^{(m)} > 0$.

En éste último caso se dice que los estados i y j *se comunican*.

Proposición 3.2.1: La relación de comunicación es una relación de equivalencia.

- 1. Reflexividad: $i \leftrightarrow i$, ya que $p_{ii}^{(0)} = \delta_{ii}$.
- 2. Simetría: Si $i \leftrightarrow j$ entonces $j \leftrightarrow i$. Es inmediato, a partir de la definición.
- 3. Transitividad: Si $i \leftrightarrow j$ y $j \leftrightarrow k$ entonces $i \leftrightarrow k$. También es aplicación inmediata de la definición.

Definición 3.2.2: Si una cadena de Markov tiene todos sus estados en la misma clase de equivalencia, se dice que es *irreducible*.

En una cadena de Markov irreducible hay una probabilidad positiva de un estado a cualquier otro estado.

Definición 3.2.3: Una cadena de Markov es *aperiódica* si el máximo común divisor de m para $P^m(i, j) > 0, \forall x$ es 1. El complementario es llamado cadena de Markov *periódica*, y en tales procesos existen estados que no comunican para ciertos números de pasos.

Definición 3.2.4: Una cadena que es a la vez *aperiódica* e *irreducible* se llama *regular*.

Definición 3.2.5: Un estado se llama transitorio si hay una probabilidad positiva de que el proceso no vuelva al estado i .

Definición 3.2.6: $f_{ii}^{(n)}$ = probabilidad de que empezando en el estado i , el proceso vuelva a i por primera vez en n pasos.

Definición 3.2.7:

$$f_{ii}^* = \sum_{n=1}^{\infty} f_{ii}^{(n)} \quad (3.9)$$

Definición 3.2.8: Un estado se llama recurrente cuando el regreso a él es el suceso seguro.

Definición 3.2.9: La esperanza matemática del número de pasos que se necesitan para volver al mismo estado se llama tiempo de recurrencia.

$$\mu_i = \sum_{n=1}^{\infty} n f_{ii}^{(n)} \quad (3.10)$$

Definición 3.2.10: Un estado es absorbente cuando $P_{ii} = 1$.

Está claro que si un estado es absorbente, entonces es recurrente.

3.3 Algoritmos Genéticos y Cadenas de Markov

Los algoritmos genéticos más sencillos y extendidos consisten en una población de n cadenas de bits de longitud l , a la que aplicamos los operadores de selección, mutación y cruce.

Sea N el número de posibles poblaciones. El número de posibles cadenas es $r = 2^l$.

Vamos a considerar el algoritmo genético como cadena de Markov, para lo cual cada estado va a estar formado por (x_1, x_2, \dots, x_r) , siendo x_i el número de soluciones de tipo i en la población en el tiempo t . Se debe cumplir que $x_1 + \dots + x_r = N$, siendo N el número de elementos de la población.

El espacio muestral será

$$S[l] = X = (x_1, x_2, \dots, x_r) : x_1 + \dots + x_r = N \text{ y } x_i \in \mathbb{N}, \forall i = 1, \dots, l \quad (3.11)$$

Proposición 3.3.1: El cardinal de $S[l]$ es $\aleph(S[l]) = \binom{N+r-1}{r-1}$, siendo $r = 2^l$.

Tab. 3.1: El número de estados N como función de l y P .

P	Longitud de la cadena (l)				
	1	2	3	4	5
1	2	4	8	16	32
2	3	10	36	136	528
3	4	20	120	816	5984
4	5	35	330	3876	52360
5	6	56	792	15504	376992
6	7	84	1716	54264	2324784
7	8	120	3432	170544	12620256
8	9	165	6435	490314	61523748
9	10	220	11440	1307504	273438880
10	11	286	19448	3268760	1121099408

Desafortunadamente, el tamaño $N \times N$ de la matriz Q para las aplicaciones típicas es inmanejable ya que el número de estados N crece rápidamente con la población P y la longitud de la cadena l .



Las poblaciones posibles vienen descritas por la matriz Z , que es una matriz $N \times r$. La i -ésima fila $\phi_i = \leq z_{i,0}, \dots, z_{i,r-1} \geq$ de Z es el vector de incidencia de la población i -ésima. En otras palabras, $z_{i,y}$ es el número de veces que aparece la cadena y en la población i -ésima, donde y es la representación entera de la cadena binaria. Por ejemplo, supongamos $l = 2$ y $P = 2$. Entonces $r = 4$, $N = 10$, y la matriz Z sería la que se muestra en la tabla 3.2

Tab. 3.2: La matriz Z para $l = 2$ y $P = 2$.

Estado	Cadena binaria			
	00	01	10	11
P1	0	0	0	2
P2	0	0	1	1
P3	0	0	2	0
P4	0	1	0	1
P5	0	1	1	0
P6	0	2	0	0
P7	1	0	0	1
P8	1	0	1	0
P9	1	1	0	0
P10	2	0	0	0

Nix y Vose definieron dos operadores matemáticos, \mathcal{F} y \mathcal{M} , donde \mathcal{F} viene determinado por la función de salud, y \mathcal{M} depende del coeficiente de mutación μ , el coeficiente de recombinación χ y de la forma de mutación y recombinación usada (en su artículo suponen que el operador de mutación es el estándar (bit-flipping) y que la recombinación se hace por un sólo punto y que produce un único descendiente). Con \mathcal{F} y \mathcal{M} definidos, es posible calcular la probabilidad de transición p_{ij} desde el estado i al estado j como

$$Q(i, j) = p_{ij} = P! \prod_{y=0}^{r-1} \frac{\mathcal{M} \left[\frac{\mathcal{F}\phi_i}{|\mathcal{F}\phi_i|} \right]_y^{z_{jy}}}{z_{jy}!} \quad (3.12)$$

Esto es, dados \mathcal{F} y \mathcal{M} , p_{ij} especifica qué tan probable es que un Algoritmo Genético en estado i pase a estado j en la siguiente generación. Se puede ver en esta ecuación que se ha supuesto que la selección se hace de manera proporcional a la salud.

Si el coeficiente de mutación μ no es cero, todos los estados tienen probabilidad no cero de ser alcanzados.

3.4 Convergencia y Comportamiento Asintótico

Se puede conservar el mejor elemento de una generación e introducirlo en la siguiente de forma automática (elitismo) o bien hacer que los operadores de mutación y cruce sean elitistas.

El operador de mutación se dice que es elitista cuando se escoge para la nueva generación el individuo con mejor salud entre el original y el mutado.

El operador cruce es elitista cuando se escoge para la nueva generación el individuo de mejor salud, entre los dos padres y los dos hijos del cruce.

El operador selección utiliza como función de probabilidad la razón entre la salud del individuo y la salud global y se suele denominar salud proporcional,

$$P(b_i) = \frac{f(b_i)}{\sum_{i=1}^n b_i}$$

donde f es una función positiva y por tanto, $P(b_i) > 0$, $\forall i = 1, \dots, N$.

Si consideramos solamente el operador selección, se puede comprobar que:

Teorema 3.4.1: Sea b^* el punto de la población con mayor salud,

$$b^* = \arg \max \{f(b_i)/b_i \in P_t\}.$$

Entonces

$$p(k) \xrightarrow{k \rightarrow \infty} (0, \dots, 0, N, 0, \dots, 0). \quad (3.13)$$

Siendo $p(k)$ =vector de distribución de probabilidad y N ocupa la posición i .

Utilizando los tres operadores, selección, mutación y cruce, en la forma usual tenemos que:

Teorema 3.4.2: Todos los estados de $S[k]$ son recurrentes.

Para comprobarlo basta darse cuenta de que mediante el operador mutación se puede pasar de un estado a otro cualesquiera, es decir, $p(X_i, X_j) > 0$.

Esto es tanto como decir que la cadena es irreducible.

Teorema 3.4.3: Consideremos una cadena de Markov X_t en el espacio de estados S , con probabilidad de transición $P(i, j)$. Si la cadena de Markov es regular entonces existe una única distribución de probabilidad $\pi(j)$, $j \in S$ tal que

$$\lim_{m \rightarrow \infty} \text{Prob}(X_m = j | X_0 = i) = \pi(j) > 0, j \in S \quad (3.14)$$

o en términos de la matriz de probabilidad a m pasos P^m

$$\lim_{m \rightarrow \infty} P^m(i, j) = \pi(j) > 0, j \in S \quad (3.15)$$

Esta distribución límite π se llama distribución estacionaria de la cadena de Markov, y $\pi(j)$ puede ser vista como la probabilidad de estar a largo plazo en el estado j .

Proposición 3.4.1: Como todos los estados son accesibles, el algoritmo no converge, aunque la distribución de probabilidad se estabiliza en el tiempo.

Podemos plantearnos qué debemos hacer para asegurar que el algoritmo converja al punto óptimo. Esto es tanto como decir que debe haber dos clases: una absorbente formada por todos los elementos de la población iguales al óptimo y la otra clase formada por todas las demás configuraciones.

Para conseguir esto basta asegurar que el mejor elemento de una generación siempre pasa a la generación siguiente, es decir es necesario introducir elitismo.

Supongamos que nuestro operador de mutación es elitista, es decir, que se muta al individuo y se escoge al de mayor salud entre el antiguo y el nuevo. Supongamos también que el operador de cruce es elitista, es decir, que cruzamos los dos elementos elegidos y escogemos para la nueva población el individuo mejor de los cuatro (los dos padres y los dos hijos).

Proposición 3.4.2: Si los operadores de mutación y cruce son elitistas, entonces la cadena de Markov está formada por una clase absorbente formada por $X = (0, 0, \dots, 0, N, 0, \dots, 0)$ donde N ocupa la posición de la solución óptima y otra clase que es transitoria formada por los demás estados.

Corolario 3.4.1: En este caso podemos asegurar que el algoritmo converge a la solución óptima.

Es decir, que si utilizamos un algoritmo genético con mutación y cruce elitistas, tenemos asegurado que el algoritmo converge al punto óptimo del espacio, ya que hay una clase absorbente correspondiente al punto óptimo y otra clase transitoria formada por todos los demás puntos.

4. MÉTODO DE OPTIMIZACIÓN GENÉTICA RESTRINGIDA (RGO)

4.1 Introducción

Los algoritmos para la optimización de funciones están limitados generalmente a funciones regulares y unimodales. Sin embargo, muchas funciones son multimodales, discontinuas y no diferenciables. Para optimizar este tipo de funciones se han empleado métodos de muestreo estocástico. Mientras que las técnicas tradicionales de búsqueda utilizan características del problema para determinar el siguiente punto muestral (por ejemplo, gradientes, Hessianos, linealidad, y continuidad), las técnicas de búsqueda estocástica no hacen tales suposiciones. En cambio el siguiente punto (o puntos) se determina basándose en reglas de muestreo y decisión estocásticas, en vez de reglas de decisión determinísticas.

Los algoritmos genéticos han sido usados para resolver problemas difíciles con funciones objetivo que no poseen buenas propiedades, tales como continuidad, diferenciabilidad, condición de Lipschitz, etc) [Davis [21], Goldberg [35], Holland [44], Michalewicz[81]]. Estos algoritmos mantienen y manipulan una familia, o población, de soluciones e implementan una “supervivencia de los mejor adaptados” como estrategia de búsqueda de mejores soluciones. Esto provee un paralelismo implícito y explícito que permite la explotación de varias regiones prometedoras al mismo tiempo. El paralelismo implícito es debido a la teoría de esquemas desarrollada por Holland, mientras que el paralelismo explícito es debido a la manipulación de una población de puntos (la evaluación de la salud de estos puntos es fácil de realizar en paralelo).

Un problema que ha sido poco estudiado en la literatura de optimización es el de la optimización dinámica, es decir de funciones que varíen con el tiempo. Este es un problema especialmente interesante para la Teoría de Control, puesto que, en mayor o menor medida, los sistemas con los que trabajamos varían con el tiempo, además de presentar los problemas anteriores.



4.2 Características de los Algoritmos Genéticos

Para realizar esta optimización de funciones objetivo que no poseen buenas propiedades y que además varían con el tiempo hemos considerado que el mejor punto de partida eran los algoritmos genéticos. Sobre estos algoritmos se pueden realizar una serie de modificaciones para la optimización de las funciones que nos interesan.

Las características principales de los algoritmos genéticos son:

- Los algoritmos genéticos no trabajan directamente con los objetos, sino con la codificación de esos objetos, que pueden ser un número, un conjunto de parámetros, etc. A esta codificación le llamamos cromosoma.
- La optimización se puede realizar con parámetros continuos o discretos.
- Los algoritmos genéticos realizan la búsqueda mediante toda una generación objetos, no buscan un único elemento.
- Los algoritmos genéticos utilizan una función objetivo que nos da la información de lo adaptados que están los individuos, y no las derivadas u otra información auxiliar.
- Las reglas de transición son probabilísticas, no determinísticas.
- Optimizan parámetros con superficies de coste extremadamente complejas.
- Pueden saltar fuera de un mínimo o máximo local.
- Trabajan con datos generados numéricamente, datos experimentales o funciones analíticas.

4.3 Optimización Genética Restringida (RGO)

En la literatura, el uso de los algoritmos genéticos como método de optimización se hace off-line y el tiempo utilizado es bastante grande.

Hemos utilizado nuevos mecanismos para usarlos como lo hace la Naturaleza: 'on-line' y de una manera local o semi-local. Hemos demostrado que los algoritmos genéticos se pueden usar como un método aceptablemente rápido de optimización de funciones que varíen con el tiempo.

Usualmente los algoritmos genéticos se usan como un método de búsqueda global y en paralelo. Se evalúan simultáneamente muchos puntos, aumentando la probabilidad de encontrar el óptimo global.

En optimización de funciones que varían con el tiempo, esta característica es interesante en las primeras generaciones, para encontrar la cuenca de atracción correcta, pero no es interesante después y además consume mucho tiempo de cómputo.

Por esta razón, es deseable un método que funcione como método de optimización global en las primeras etapas, para encontrar la cuenca de atracción correcta y que posteriormente se comporte como un método de optimización local o semi-local.

Sería deseable también que tuviera propiedades parecidas a las de los métodos basados en el gradiente, pero sin los inconvenientes de que sea necesaria la existencia de derivadas y su sensibilidad al ruido. Los algoritmos para la optimización de funciones están generalmente limitados a funciones regulares convexas. Sin embargo muchas funciones son multi-modales, discontinuas, y no diferenciables. Mientras que los métodos tradicionales de búsqueda utilizan características del problema para determinar el siguiente punto (gradientes, hessianos, linealidad, y continuidad) la búsqueda genética no hace estas suposiciones.

Cuando queremos optimizar funciones que varían con el tiempo las cosas se complican aún más porque necesitamos un método que sea lo suficientemente rápido para seguir las variaciones de los puntos óptimos de la función.

4.3.1 RGO

El método de búsqueda restringida consiste en hacer la búsqueda en un entorno del punto correspondiente al último modelo identificado y tomar el punto mejor adaptado como centro del entorno de búsqueda de la nueva generación. A continuación se vuelve a tomar el individuo con mejor salud como centro del nuevo entorno de búsqueda y así sucesivamente.

La codificación de cada cromosoma corresponde al vector diferencia con respecto al centro del entorno. Éste punto también se introduce en la siguiente generación (es la cadena formada por ceros). El punto correspondiente al mejor de cada generación y que se va a guardar como centro del entorno de la generación siguiente se guarda aparte de la generación y representando coordenadas absolutas..

Para evaluar la función de salud, consideramos el centro del entorno más la decodificación de cada uno de los cromosomas. Es decir, el método trabaja de un modo incremental. Este comportamiento pretende simular el método del gradiente, pero sin utilizar derivadas y puede ser utilizado cuando las señales tienen mucho ruido. Vamos teniendo nuevas generaciones orientadas en la dirección de mayor decrecimiento de la función de coste, y con una distancia al centro próxima

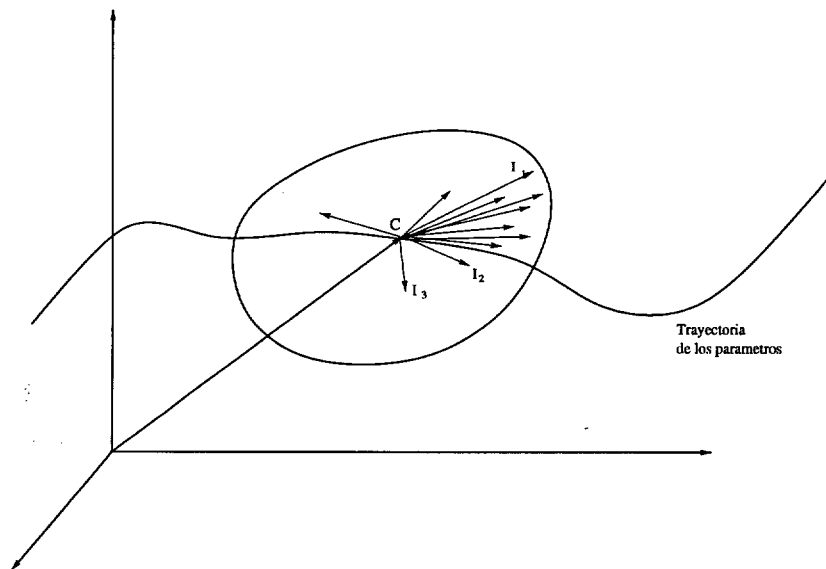


Fig. 4.1: Mecanismo de búsqueda de la Optimización Genética Restringida.

a la correspondiente a la velocidad de cambio del sistema.

Podemos hacer una búsqueda en un entorno grande al principio y después reducir su radio. De hecho se realiza una búsqueda global al principio y local al final. Este método reduce la probabilidad de encontrar un mínimo local.

El radio se ha tomado proporcional a la incertidumbre de la estimación con extremos superior e inferior. La manera de calcular la incertidumbre depende de la aplicación que se hace de la optimización, por ejemplo, en el caso de la estimación de estados, usamos la distancia de Mahalanobis 6.43

$$d = (\hat{x}(k|k-1) - \hat{x}(k-1|k-1))' P^{-1}(k|k) (\hat{x}(k|k-1) - \hat{x}(k-1|k-1)) \quad (4.1)$$

que mide la incertidumbre de la estimación $\hat{x}(k)$. En el caso de la identificación de sistemas el radio se ha tomado proporcional al criterio MDL (Minimum Description Length) con extremos superior e inferior. También se pueden utilizar otros criterios como el AIC (Akaike's Information theoretic Criterion) o el FPE (Final Prediction-Error criterion)(ver Ljung[69]).

En nuestro algoritmo hemos implementado los siguientes operadores: reproducción, cruce, mutación, elitismo, inmigración, ranking y búsqueda restringida.

Hemos utilizado el mecanismo de ranking para regular el número de descendientes que un individuo puede tener, porque si un individuo es muy bueno, puede tener demasiados descendientes y hacer que la diversidad genética descienda mucho.

En caso de que el sistema vaya variando rápidamente en el tiempo, podemos incluir los mecanismos de reescalado, dominancia trialélica e inmigración.

Este mecanismo se ha implementado utilizando como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_{n-k})^2$ es la función de pérdida (loss function) y "cte" es una constante suficientemente grande (10^8).

Es importante destacar que el método RGO realiza la búsqueda con más énfasis en una dirección preferente, la dirección de máxima pendiente y a una distancia preferente. Este hecho permite reducir la búsqueda de 50 ó 100 dimensiones a 1 ó 2. Por ésta razón éste método es comparable al método del gradiente en precisión y velocidad, con la ventaja de que es aplicable cuando las expresiones no son diferenciables y las señales contienen ruido.

4.4 Algoritmo

1. Se obtiene un primer conjunto de datos.
2. Se crea la población inicial para deducir el tipo y las características del sistema.
3. Se evalúa la salud de los individuos.
4. Se hallan las características del sistema, por ejemplo, en el caso de identificación, nos interesan el tiempo de retardo y el orden, o también nos puede interesar la elección de modelo.
5. Se recogen datos $u(k)$, $y(k)$.
6. Se obtiene una primera estimación del modelo utilizando mínimos cuadrados extendidos o alguna otra técnica.
7. Hacemos $k = 1$
8. Se crea la población inicial, tomando los individuos como diferencia respecto al modelo anterior.
9. Se evalúa la salud de los individuos.
10. Comienzo del bucle.
11. Se aplican los mecanismos de selección, cruce y mutación.
12. Se clona el mejor individuo de la generación anterior.

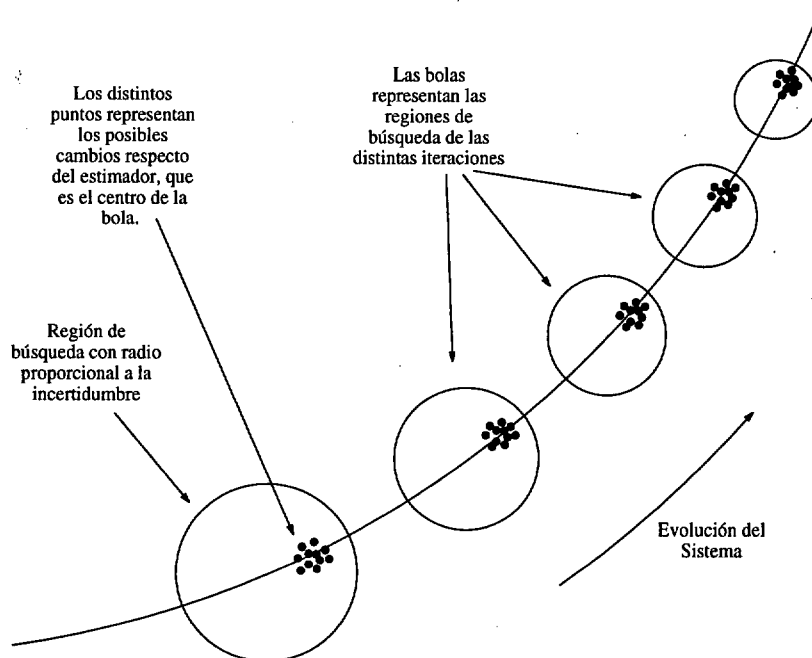


Fig. 4.2: Mecanismo de seguimiento de la Optimización Genética Restringida.

13. Aplicamos el mecanismo de inmigración.
14. Se evalúa la salud de los individuos de la nueva generación.
15. Seleccionamos el individuo que optimiza el criterio deseado

$$\hat{\theta}^k = \arg \min_{\theta} V_N(\theta). \quad (4.2)$$

16. Sumamos el mejor individuo al modelo previo.
17. Calculamos la incertidumbre para utilizarla en el cálculo del nuevo radio de la región de búsqueda.
18. Se recogen datos $u(k)$, $y(k)$.
19. Hacemos $k = k + 1$ y repetimos los pasos 10-19.

4.5 Características de la RGO

- Se pueden utilizar como método de optimización en una bola de radio que vaya variando según la incertidumbre de las estimaciones, con lo que evitamos buscar en una región excesivamente amplia como hacen los métodos de búsqueda global y evitamos la posibilidad de caer en un óptimo local como hacen los métodos de optimización local.
- Éste método está especialmente pensado para utilizarlo en la optimización de funciones variantes en el tiempo. Lo que permite usarlo en la identificación de sistemas, ya sea en la forma de la función de transferencia o en variables de estado y para la estimación de estados por medio de un filtro de Kalman Generalizado.

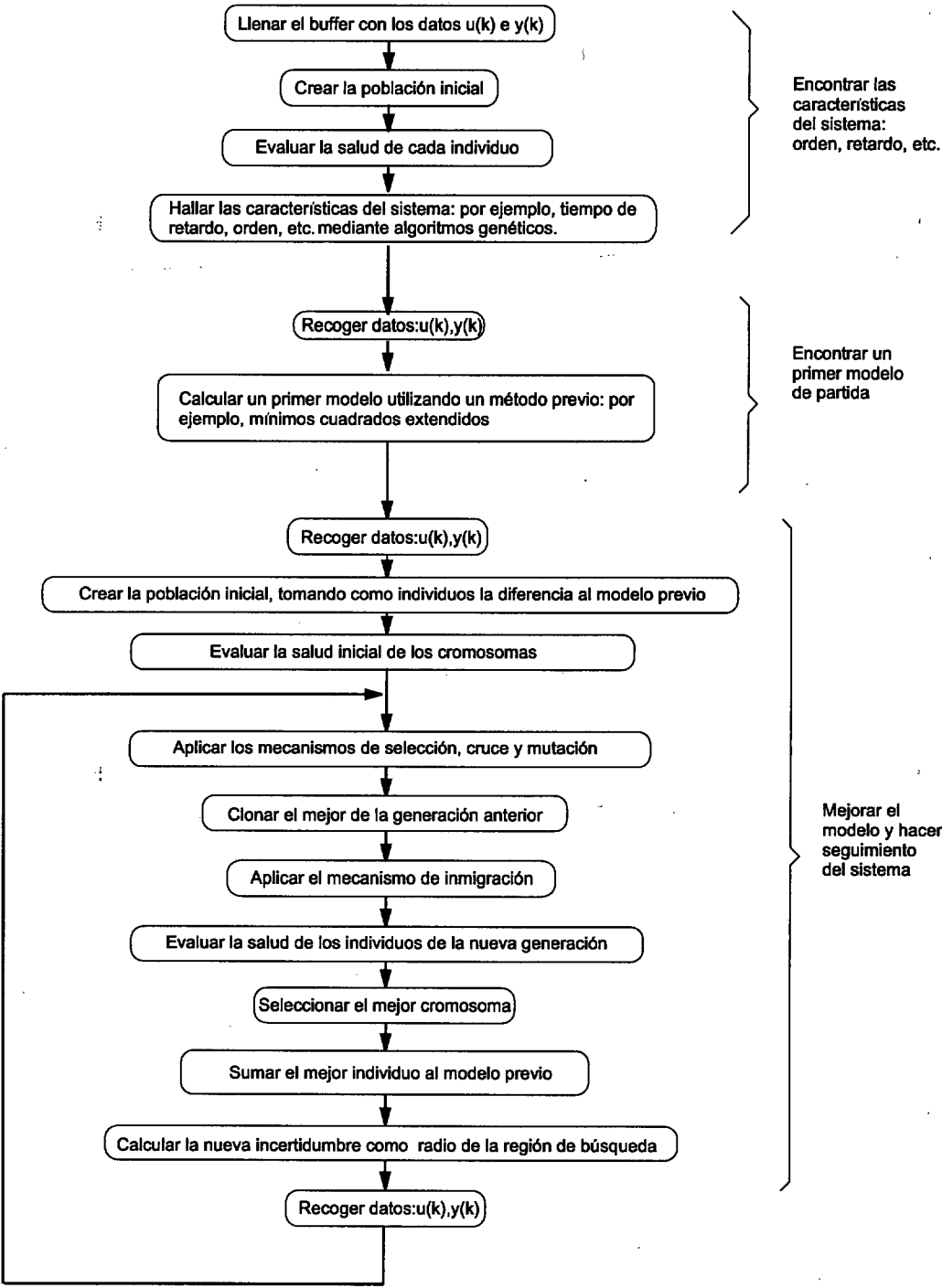


Fig. 4.3: Diagrama de flujo de la Optimización Genética Restringida RGO

5. IDENTIFICACIÓN DE SISTEMAS NO LINEALES

5.1 Introducción

La identificación de sistemas no lineales mediante modelos lineales o no lineales es un trabajo difícil y complicado. La mayoría de las veces debemos partir de los conocimientos humanos sobre la dinámica del sistema. La identificación posterior de los parámetros la realizamos a mano, utilizando las propiedades heurísticas de cada familia de modelos.

Los métodos más usuales en la identificación de sistemas están basados en mínimos cuadrados o en el método de máxima verosimilitud que son métodos locales guiados por el gradiente y necesitan una función de pérdidas (loss function) diferenciable y un espacio de búsqueda suave. A menudo fallan en la búsqueda de un óptimo global cuando el funcional no es diferenciable o es no lineal en sus parámetros.

Esto tiene el inconveniente de que si las señales tienen ruido coloreado, como suele ser habitual en la práctica, los métodos basados en mínimos cuadrados tienen el fracaso asegurado.

Si además estamos implementado el método en línea el ruido estar correlacionado con los regresores y el método explotar, fenómeno conocido como blow-up.

Para evitar esto en algunos casos sencillos, se puede usar un factor de olvido muy próximo a la unidad, pero en la mayoría de los casos de sistemas no lineales variantes en el tiempo esta técnica falla.

A menudo fallan en la búsqueda de óptimos globales cuando éste espacio de búsqueda es no diferenciable, o no lineal en sus parámetros. Los métodos de optimización basados en algoritmos genéticos pueden mejorar los resultados porque no utilizan las propiedades de diferenciabilidad o de ser lineal en los parámetros.

En 1992, Kristinsson y Dumont [64] hicieron un primer estudio de la identificación de sistemas y del control off-line con algoritmos genéticos de sistemas dinámicos continuos y discretos. Comprobaron que los algoritmos genéticos son efectivos en ambos dominios y que es posible identificar directamente parámetros

físicos, o bien, polos y ceros. Hicieron simulaciones con sistemas de fase mínima y no mínima y de un sistema con dinámica no modelada.

En 1993, Flockton y White [25] utilizan los algoritmos genéticos para hacer otra identificación off-line de polos y ceros con algunas pequeñas mejoras.

Posteriormente, en 1995, Tan, Li y Murray-Smith [99] usan los algoritmos genéticos con "simulated annealing" para hacer una identificación off-line y la linealización de sistemas no lineales, invariantes del tiempo con un modelo ARMAX. Utiliza esta técnica de búsqueda para identificar los parámetros de un sistema descrito por un modelo ARMAX en variables de estado, en presencia de ruido blanco y para aproximar un sistema no lineal multivariable por un modelo en el espacio de estados, lineal e invariante en el tiempo.

En este mismo año, Iba, deGaris y Sato [50] usan una técnica numérica off-line, que integra una búsqueda adaptativa de una estructura de árbol basada en programación genética y un mecanismo de ajuste local de parámetros empleando búsqueda estadística. En la programación genética tradicional, la recombinación causa frecuentes rupturas de los bloques de construcción y la mutación puede causar abruptos cambios en la semántica. Para superar estas dificultades ellos implementan la programación genética tradicional con una búsqueda local "hill-climbing", usando un procedimiento de ajuste de parámetros.

Todos estos métodos trabajan off-line y precisan que el sistema sea invariante del tiempo. El último además utiliza un método no paramétrico.

Este capítulo describe un método de identificación on-line de sistemas no lineales, variantes en el tiempo que tiene altas prestaciones. Hemos utilizado modelos OE, ARMAX y NARMAX como semillas para el método RGO y se puede extender a otros modelos similares, como Box-Jenkins, el Lineal General, Modelos de Polos y Ceros o la descripción en el Espacio de Estados, utilizando para ello optimización genética restringida.

Las principales diferencias del RGO con los anteriores métodos es que éste trabaja en línea, identifica modelos lineales y no lineales, variantes en el tiempo y usa modelos paramétricos convencionales.

5.2 Implementación del Algoritmo

El método realiza la búsqueda en un entorno del punto correspondiente al último modelo identificado y toma el punto mejor adaptado como centro del entorno de búsqueda de la nueva generación. Este último punto se guarda aparte de la generación. La codificación de cada cromosoma corresponde al incremento con respecto al centro del entorno.

Para evaluar la función de salud, consideramos el centro del entorno más la decodificación de cada uno de los cromosomas. Es decir, el método trabaja de un modo incremental. Este comportamiento simula el método del gradiente sin utilizar derivadas y puede ser utilizado cuando las señales tienen mucho ruido.

Vamos teniendo nuevas generaciones orientadas en la dirección de mayor decrecimiento de la función de coste, y con una distancia al centro próxima a la correspondiente a la velocidad de cambio del sistema.

Podemos hacer una búsqueda en un entorno grande al principio y después reducir su radio. De hecho se realiza una búsqueda global al principio y local al final. Este método reduce la probabilidad de encontrar un mínimo local.

El radio se ha tomado proporcional al criterio MDL (Minimum Description Length) con extremos superior e inferior. También se pueden utilizar otros criterios como el AIC (Akaike's information theoretic criterion) o el FPE (Final prediction-error criterion)(ver [69]).

En nuestro algoritmo hemos implementado los siguientes operadores: reproducción, cruce, mutación, elitismo, inmigración, ranking y búsqueda restringida.

Hemos utilizado el mecanismo de ranking para regular el número de descendientes que un individuo puede tener, porque si un individuo es muy bueno, puede tener demasiados descendientes y hacer que la diversidad genética descienda mucho.

Este mecanismo se ha implementado utilizando como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_{n-k})^2$ es la función de pérdida (loss function) y "cte" es una constante grande (10^8).

5.3 Identificación multimodelo y multiparámetro de sistemas no lineales mediante modelos paramétricos lineales.

5.3.1 Identificación con modelos lineales

Un modelo de entrada-salida es un modo de describir un sistema dinámico basándose exclusivamente en la relación entre las entradas y salidas del sistema.

Algunos de los modelos lineales mas conocidos son:

ARX (Auto-Regressive with eXogenous inputs), cuya expresión es

$$A(q-1)y(t) = B(q-1)u(t - nk) + e_k \quad (5.1)$$

, que puede ser visto como una manera de determinar el siguiente valor de la salida dadas las observaciones anteriores y las entradas. Aquí $A(q^{-1})$ y $B(q^{-1})$

son polinomios en el operador desplazamiento hacia atrás q^{-1} e $y(t)$, $u(t)$, y $e(t)$ son las salidas, entradas y ruido, respectivamente. El ruido $e(t)$ es una sucesión aleatoria normalmente distribuida de media cero y varianza σ^2 .

OE (Output Error) cuya expresión es

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t - nk) + e(t) \quad (5.2)$$

con la fuente de error $e(t)$ que es la diferencia (error) entre la salida real y la salida libre de ruido (teórica).

ARMAX (Auto-Regressive Moving Average with eXogenous inputs), cuya expresión es

$$A(q^{-1})y(t) = B(q^{-1})u(t - nk) + C(q^{-1})e(t) \quad (5.3)$$

. En éste modelo la expresión $A(q^{-1})y(t) = e(t)$ representa la auto-regresión, $y(t) = C(q^{-1})e(t)$ representa la media móvil de ruido blanco, mientras que $B(q^{-1})u(t)$ representa la entrada externa. Otros modelos pueden ser usados, como por ejemplo la representación en el espacio de estados o el modelo de ceros y polos.

5.3.2 Identificación RGO para modelos lineales

La entrada excitadora $u(t)$ se ha escogido como una secuencia binaria pseudo-aleatoria (PRBNS), con ruido coloreado.

Como primer paso, para hacer esta identificación paramétrica, realizamos una primera identificación de múltiples modelos de tipo ARX, $A(q^{-1})y_k = z^{-d}B(q^{-1})u_k + e(t)$, mediante el método de mínimos cuadrados y por medio de los algoritmos genéticos decidimos qué grados de los polinomios numerador y denominador y qué retardo son más convenientes para nuestros propósitos.

Es decir, queremos un modelo que se ajuste lo más posible al sistema, pero que no tenga unos grados innecesariamente grandes, para ello se pueden utilizar los criterios MDL, AIC o FPE. En particular nosotros hemos utilizado el criterio MDL.

En un segundo paso, calculamos un modelo, OE $y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t - nk) + e(t)$ y ARMAX, $A(q^{-1})y_k = q^{-d}B(q^{-1})u_k + C(q^{-1})e(t)\omega_k$, por medio de regresión pseudo-lineal. Este modelo todavía no se ajusta lo suficiente porque se ha supuesto en el cálculo que el sistema es lineal, cosa que en general no es cierta.

Para mejorar esta estimación realizamos una búsqueda de modelos ARMAX, $A(q^{-1})y_k = q^{-d}B(q^{-1})u_k + C(q^{-1})e_k$, cuyos coeficientes sean próximos al calculado anteriormente y que tengan una función de pérdida lo más pequeña posible.

Esta búsqueda la realizamos por medio de un algoritmo genético. En paralelo, vamos haciendo una búsqueda de modelos OE , $y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t - nk) + e(t)$, utilizando la misma técnica.

Para hacer la función de pérdida $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_{n-k})^2$ lo más pequeña posible, se ha utilizado como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde "cte" es una constante suficientemente grande (10^8) para mantener la diversidad de la población. La probabilidad de reproducción se ha calculado como $F_n(t) = F(t)/\bar{F}$.

En cada generación, sustituimos el modelo anterior por el mejor de la nueva generación como centro del entorno donde se realiza la búsqueda, de esta manera, en caso de un sistema LTI, el mejor modelo debe ir tendiendo al valor que minimiza el funcional y la población irá tendiendo a cero. A continuación realizamos una nueva lectura de la entrada y la salida del sistema.

De esta manera conseguimos un conjunto de aproximaciones sucesivas a mejores modelos en caso de el sistema sea invariante en el tiempo y si el sistema es variable, pero de variación lo suficientemente lenta, entonces el modelo va siguiendo al sistema en sus variaciones.

Podemos además aprovechar las características convenientes de la convergencia global si al principio tomamos una bola lo suficientemente grande, de manera que reduzcamos la probabilidad de quedarnos en un mínimo local. Una vez que sepamos que seguimos al mínimo adecuado, podemos reducir el tamaño de la bola y al mismo tiempo de la población, para conseguir una mayor velocidad de ejecución del algoritmo.

Para que el algoritmo no tenga una convergencia prematura y se homogeneice demasiado la población, podemos aumentar el parámetro de probabilidad de mutación.

Aunque este algoritmo es costoso en cuanto a operaciones, es lo suficientemente rápido como para ser empleado en línea, ya que el cálculo de un modelo con 35 coeficientes tarda una décima de segundo en un PC (120 MHz). A esto hay que añadir que las aplicaciones a que típicamente se desea aplicar este identificador, es decir a los sistemas de control adaptativo, poseen dos bucles, uno rápido, que es el de control, y otro lento que es el identificador. Por tanto se puede aplicar a sistemas con una constante de tiempo de orden de las centésimas de segundo.

Debemos darnos cuenta de que los algoritmos genéticos hacen una búsqueda preferentemente en una dirección, la de máximo decrecimiento, mientras que el método Montecarlo la hace con igual densidad en todas las direcciones, con lo que si estamos buscando en un espacio de 50 a 100 dimensiones, es evidentemente superior.

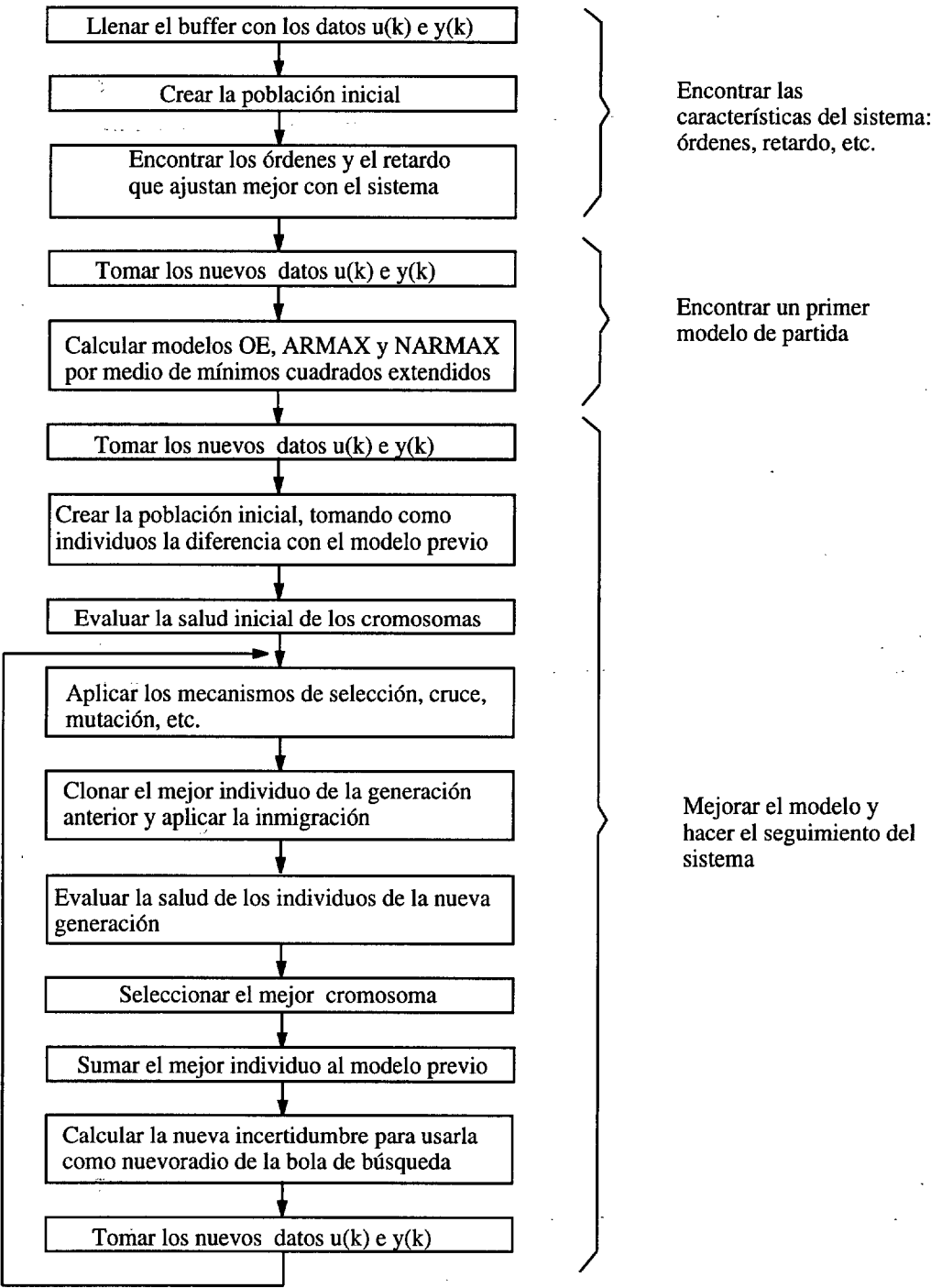


Fig. 5.1: Diagrama de flujo de la RGO.

5.4 Identificación multimodelo y multiparámetro de sistemas no lineales mediante modelos paramétricos no lineales.

Usualmente, modelamos el sistema basándonos en el conocimiento físico del sistema y después intentamos calcular los parámetros desconocidos a mano. Éste método tiene dos problemas: necesitamos conocimiento previo del sistema, y si las señales tienen ruido coloreado, los métodos basados en mínimos cuadrados no funcionan satisfactoriamente.

5.4.1 Modelos NARMAX

Entre los modelos no lineales, uno de los mas interesantes, es el modelo NARMAX (Non-linear Auto-Regressive Moving Average with eXogenous inputs), que puede ser descrito como

$$y(t) = F[y(t-1), \dots, y(t-n_y), u(t-d), \dots, u(t-d-n_u+1), \varepsilon(t-1), \dots, \varepsilon(t-n_\varepsilon)] + \varepsilon(t) \quad (5.4)$$

donde t representa el tiempo discreto, $y(t)$, $u(t)$ y $\varepsilon(t)$ son la salida, la entrada, y el error de predicción, respectivamente, n_y , n_u y n_ε son los órdenes correspondientes, $F[\cdot]$ es una función no lineal y dd es el mínimo retardo de la entrada [Leontaritis 85] El modelo polinomial NARMAX tiene como expresión

$$y(k) = a_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_y} \sum_{j=1}^{n_u} c_{ij} y(k-i) u(k-j) \quad (5.5)$$

nos permite una descripción buena y sencilla de una amplia clase de sistemas no lineales.

Por ejemplo, si el modelo exacto NARMAX es

$$y(k) = y(k-1)e^{-u(k-1)} \quad (5.6)$$

podemos desarrollar en serie la exponencial

$$\exp(-x) = \sum_{i=0}^{\infty} \frac{(-x)^i}{i!} \quad (5.7)$$

y emplear un modelo polinomial aproximado NARMAX

$$y(k) = y(k-1) - y(k-1)u(k-1) + \frac{1}{2}y(k-1)u^2(k-1) - \frac{1}{6}y(k-1)u^3(k-1) \quad (5.8)$$

Podemos comprobar que éste modelo tiene obvias ventajas sobre las series funcionales tales como las series de Volterra.



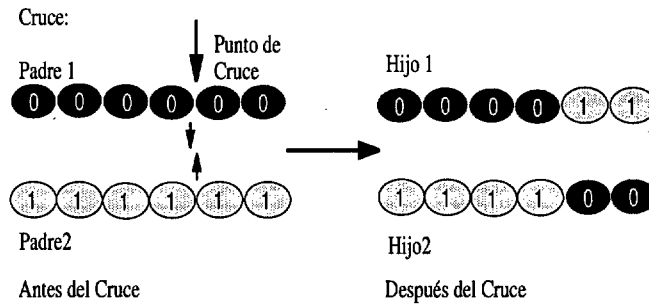


Fig. 5.2: Mecanismo de cruce

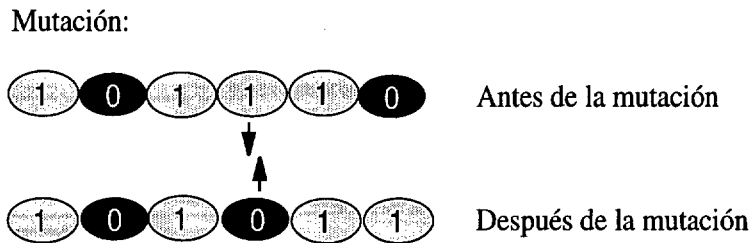


Fig. 5.3: Mecanismo de mutación

5.4.2 Identificación RGO con modelos no lineales

La entrada excitadora $u(t)$ se ha escogido como una secuencia binaria pseudo-aleatoria (PRBNS), con ruido coloreado.

Como primer paso, para hacer esta identificación paramétrica, realizamos una primera identificación de múltiples modelos de tipo ARX, $A(q^{-1})y_k = z^{-d}B(q^{-1})u_k + e(t)$, mediante el método de mínimos cuadrados y por medio de los algoritmos genéticos decidimos qué grados de los polinomios numerador y denominador y qué retardo son más convenientes para nuestros propósitos.

Es decir, queremos un modelo que se ajuste lo mas posible al sistema, pero que no tenga unos grados innecesariamente grandes, para ello se pueden utilizar los criterios MDL, AIC o FPE. En particular nosotros hemos utilizado el criterio MDL (Minimum Description Length)(ver Ljung[69] para más detalles).

En un segundo paso, calculamos un modelo ARMAX, $A(q^{-1})y_k = q^{-d}B(q^{-1})u_k + C(q^{-1})e(t)\omega_k$, por medio de regresión seudo-lineal. Este modelo todavía no se ajusta lo suficiente porque el sistema es no lineal y además la señal de salida contiene ruido coloreado y estos métodos solamente son buenos con sistemas lineales con ruido blanco.

Para mejorar esta estimación realizamos una búsqueda de modelos NARMAX,

$$y(k) = a_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_y} \sum_{j=1}^{n_u} c_{ij} y(k-i) u(k-j) \quad (5.9)$$

cuyos coeficientes sean próximos al calculado anteriormente y que tengan una función de pérdida lo más pequeña posible.

En cada generación, el último modelo es sustituido por el mejor de la nueva generación como centro del entorno donde se realiza la búsqueda. De esta manera nuestro modelo va siguiendo al modelo real. En éste entorno los individuos de las nuevas generaciones estarán orientados principalmente en la dirección de máxima pendiente de la función de pérdida (usando el criterio MDL, o bien AIC o FPE) y a la distancia que corresponde a la velocidad de cambio del sistema real.

Para hacer la función de pérdida $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_n - k)^2$ lo más pequeña posible, se ha utilizado como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde "cte" es una constante suficientemente grande (10^8) para mantener la diversidad de la población. La probabilidad de reproducción se ha calculado como $F_n(t) = F(t)/\bar{F}$.

El número de coeficientes del modelo NARMAX

$$y(k) = a_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_y} \sum_{j=1}^{n_u} c_{ij} y(k-i) u(k-j) \quad (5.10)$$

crece exponencialmente con el orden de desarrollo. Por este motivo, se ha utilizado una expansión de orden dos, con sistemas de orden menor que ocho y con cinco coeficientes para modelar el ruido.

5.5 Restricciones

Un problema del que adolecen la mayoría de los métodos de identificación es no permiten introducir fácilmente el conocimiento previo que tenemos del sistema. En la mayoría de los casos sólo se puede elegir el tipo de modelo (ARX, OE, ARMAX, BJ, etc.) procurando además que sea lineal porque de lo contrario no se pueden aplicar los métodos habituales.

Sin embargo, en el método de algoritmos genéticos lo que hacemos es optimizar numéricamente una función

$$V = \sum_{k=0}^n (y(t-k) - \hat{y}(t-k))^2 \quad (5.11)$$

y no nos cuesta ningún trabajo añadir restricciones duras de desigualdad, como $a_1 > 0$, asignando salud cero a los individuos que estén en la región factible. Las

restricciones blandas se pueden tratar, añadiendo un término de penalización al funcional V . Por ejemplo, si deseamos introducir la restricción $a_1 > 0$ como una restricción blanda, podemos añadir a la función de pérdida el término a_1^2 , cuando $a_1 > 0$.

5.6 Resultados

Para poder contrastar los resultados con otros métodos, vamos a utilizar como índices de funcionamiento los siguientes: Error RMS:

$$RMS\ error = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (5.12)$$

Error relativo RMS:

$$Relative\ RMS\ error = \frac{\sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}}{\sqrt{\frac{\sum_{i=1}^N (y_i)^2}{N}}} \times 100\% \quad (5.13)$$

Estudiaremos los resultados obtenidos al aplicar nuestro método con modelos OE, ARMAX y NARMAX a varios sistemas no lineales:

Planta 1:

$$y(k) - 0.6y(k-1) + 0.4y(k-2) = (u(k) - 0.3u(k-1) + 0.05u(k-2))\sin(0.01k) + e(k) - 0.6e(k-1) + 0.4e(k-2) \quad (5.14)$$

con condiciones iniciales nulas.

Esta planta corresponde a un sistema lineal

$$y(k) = \frac{1 - 0.3q^{-1} + 0.5q^{-2}}{1 - 0.5q^{-1}}u(k) + \frac{1 - 0.6q^{-1} + 0.4q^{-2}}{1 - 0.5q^{-1}}e(k) \quad (5.15)$$

con una pequeña oscilación sinusoidal en la ganancia.

Planta 2:

Esta planta consiste en un primer bloque lineal:

$$y(k) - 0.6y(k-1) + 0.4y(k-2) = u(k) - 0.3u(k-1) + 0.05u(k-2) + e(k) - 0.6e(k-1) + 0.4e(k-2) \quad (5.16)$$

puesto en serie con una zona muerta de 0.1, con condiciones iniciales nulas.

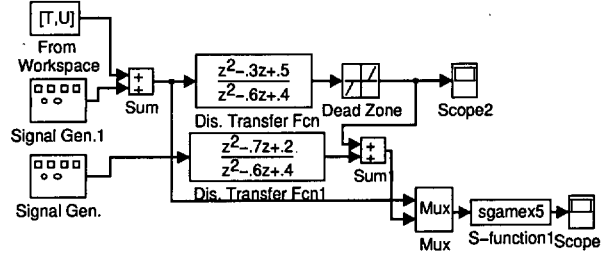


Fig. 5.4: Montaje de la Planta 2.

Planta 3:

$$y(k) = A_1 + 0.02 \sin(0.1k)y(k-1) + A_2 y(k-2) + A_3 y(k-3) + B_1 u(k-1) + B_2 u(k-2) + B_3 u(k-3) \quad (5.17)$$

con $A_1 = 2.627771$, $A_2 = -2.333261$, $A_3 = 0.697676$, $B_1 = 0.017203$, $B_2 = -0.030862$ y $B_3 = 0.014086$.

Planta 4:

$$y(k) = (0.8 - 0.5e^{-y(k-1)})y(k-1) - (0.3 + 0.9e^{y^2(k-1)})y(k-2) + 0.1 \sin(\pi y(k-1)) + u(k) \quad (5.18)$$

con condiciones iniciales nulas.

Nota: Las plantas 3 (sin la no-linealidad) y 4 han sido elegidas por haber sido utilizadas en el proyecto Psycho de identificación y control con redes neuronales.

El montaje es el mismo para todas las plantas, tenemos una entrada al sistema principal consistente en una señal *prbns* de amplitud 1. A la salida de éste le sumamos un ruido coloreado que es la salida que obtenemos al filtrar un ruido blanco de pico 0.1 con el filtro

$$\frac{1 + 0.7q^{-1} + 0.2q^{-2}}{1 - 0.6q^{-1} + 0.4q^{-2}} \quad (5.19)$$

La entrada *prbns* al sistema principal y la salida del sistema total las llevamos al bloque de identificación, que nos va dando los coeficientes del sistema para los modelos OE, ARMAX y NARMAX, así como los errores RMS (absoluto y relativo) y los índices AIC, FPE y MDL.

Los resultados obtenidos se pueden resumir en la siguiente tabla:

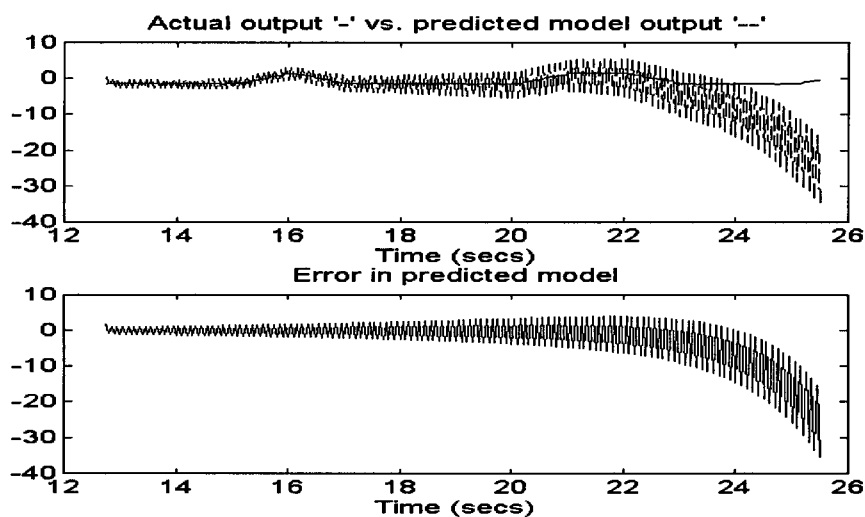


Fig. 5.5: Fenómeno de blow-up al intentar identificar la planta 1 con el bloque ARMAX de Simulink. Este efecto se debe a que se correlan los errores $\varepsilon(t)$ y $\eta(t)$ con la entrada y hace imposible la identificación de este sistema mediante este método.

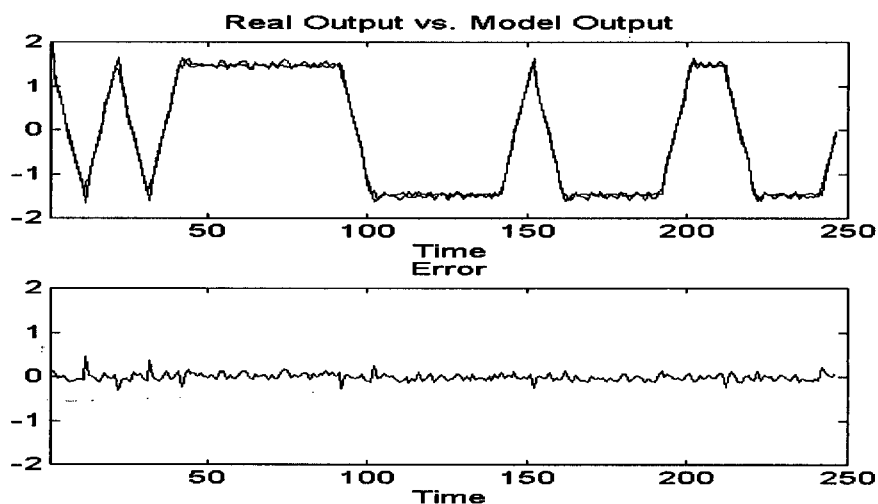


Fig. 5.6: Resultados de la identificación de la planta 1 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido.

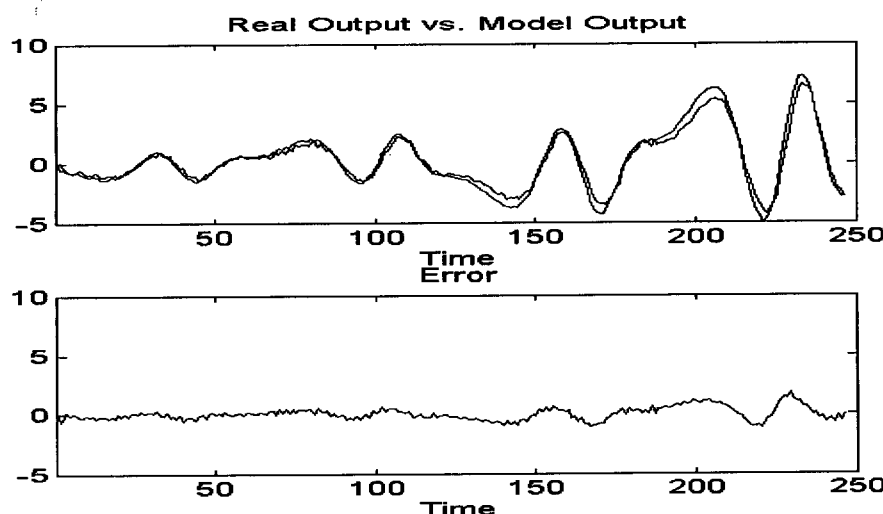


Fig. 5.7: Resultados de la identificación de la planta 2 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido.

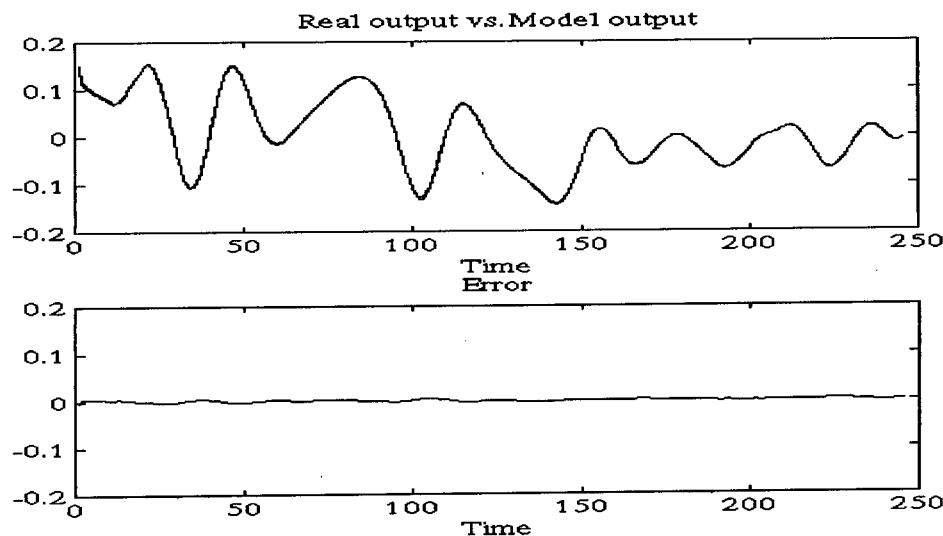
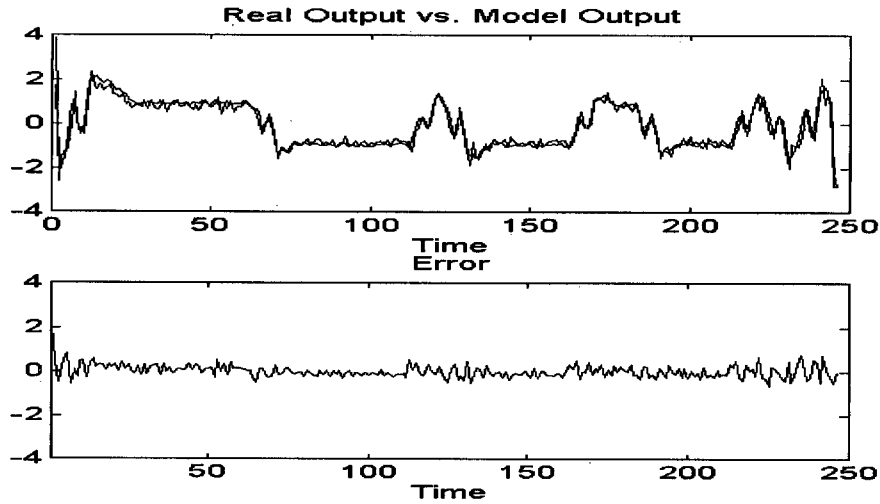


Fig. 5.8: Resultados de la identificación de la planta 3 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido.

Tab. 5.1: Comparación de los métodos de identificación de sistemas.

Error RMS				
	Planta 1	Planta 2	Planta 3	Planta 4
ARMAX Simulink	0.050	1.330	209.200	10.780
OE RGO	0.050	0.030	0.009	0.050
ARMAX RGO	0.018	0.010	0.0008	0.050
NARMAX RGO	0.017	0.003	0.0003	0.040

**Fig. 5.9:** Resultados de la identificación de la planta 4 con el método RGO. Arriba: salida del modelo vs. salida real. Abajo: error cometido.

5.7 Comparación con las Redes Neuronales

Para comparar los resultados de la identificación del método RGO-Narmax con las Redes Neuronales hemos utilizado una Red Feedforward de tipo Perceptrón Multicapa que usa como mecanismo de aprendizaje Backpropagation (que usa la regla Delta generalizada como algoritmo de entrenamiento) y con la sigmoide como función de activación. La red consta de una capa de entrada (formada por 3 neuronas: entrada, salida-1 y salida-2), 16 neuronas en la capa oculta y capa de salida (con 1 neurona, que representa la salida estimada del sistema).

En las redes Feedforward la activación de la salida b se calcula de acuerdo a la ecuación

$$b = F\left(\sum_{i=1}^N \omega_i a_i + \theta\right) \quad (5.20)$$

donde a_i son los niveles de activación de las entradas, ω_i el peso de conexión de la entrada i -ésima al elemento de salida, θ es un valor umbral y F es la función de activación, en éste caso la sigmoide $F(x) = \frac{1}{1+e^{-x}}$. El mecanismo de aprendizaje Backpropagation consiste en propagar hacia atrás, partiendo de la salida de las señales de error para calcular los ajustes que se deben aplicar a los pesos de todas las conexiones de la red. Este mecanismo es utilizado para ajustar los pesos y sesgos de las redes neuronales para minimizar la suma cuadrática de los errores de la red. Esto se realiza cambiando continuamente los valores de los pesos de la red y los sesgos en la dirección de máximo decrecimiento con respecto al error (regla Delta).

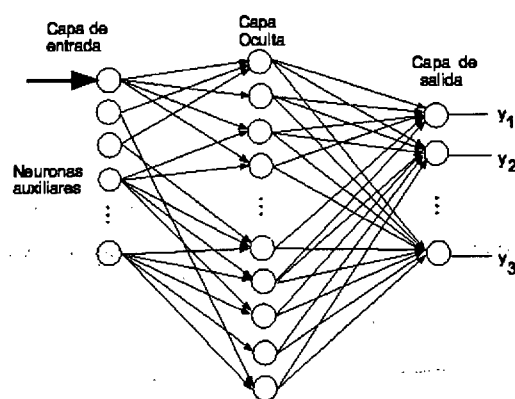


Fig. 5.10: Estructura de la Red Multicapa utilizada en la comparación.

La regla Delta es el algoritmo de entrenamiento del perceptrón para entradas y salidas continuas y se basa en que el ajuste del peso ω_i sea proporcional a la derivada, respecto del peso, del error cuadrático medio E medido en el patrón actual p ,

$$\Delta_p \omega_i = -\gamma \frac{\partial E^p}{\partial \omega_i}. \quad (5.21)$$

Desarrollando esta expresión, se obtiene,

$$\Delta_p \omega_i = \gamma \delta^p x_i \quad (5.22)$$

donde $\delta^p = d^p - o^p$ es la diferencia entre la salida deseada d^p y la salida actual o^p para el patrón p . La regla de actualización es

$$\omega_i = \omega_{i,old} + \Delta_p \omega_i. \quad (5.23)$$

Las redes Backpropagation entrenadas tienden a dar respuestas razonables cuando se le presentan entradas que no han visto nunca. Típicamente, una nueva entrada debe llevar al sistema a una salida similar a la correcta para vectores de entrada parecidos a los utilizados en el entrenamiento.

El montaje es el mismo para todas las plantas, tenemos una entrada al sistema principal consistente en una señal prbns de amplitud 1 mas una onda cuadrada de amplitud 0.4 y frecuencia 0.1. A la salida de éste le sumamos un ruido coloreado que es la salida que obtenemos al filtrar un ruido blanco de pico 0.1 con el filtro

$$\frac{z^2 + 0.7z + 0.2}{z^2 - 0.6z + 0.4} \quad (5.24)$$

Los principales problemas observados en las redes son: la fuerte dependencia de las señales utilizadas para el entrenamiento y que los sistemas deben ser invariantes en el tiempo o, a lo sumo, con variaciones pequeñas. Para poder utilizarlas en control adaptativo es necesario que la red sea capaz de aprender rápidamente en línea.

Otro problema de las Redes Neuronales es que no dan un modelo explícito del sistema.

Los resultados obtenidos se pueden resumir en la siguiente tabla:

Tab. 5.2: Comparación de los métodos de identificación: RGO y Redes Neuronales.

Error RMS				
	Planta1	Planta 2	Planta 3	Planta 4
NARMAX RGO	0.0293	0.02284	0.02201	0.1678
Red Neuronal	0.0335	0.04419	0.07069	0.407

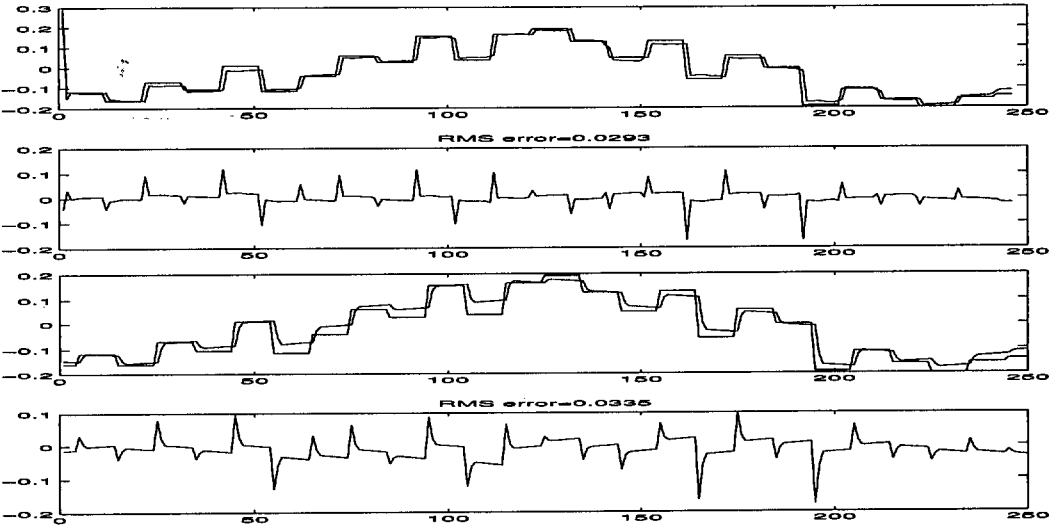


Fig. 5.11: Resultados de la identificación de la planta 1. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.

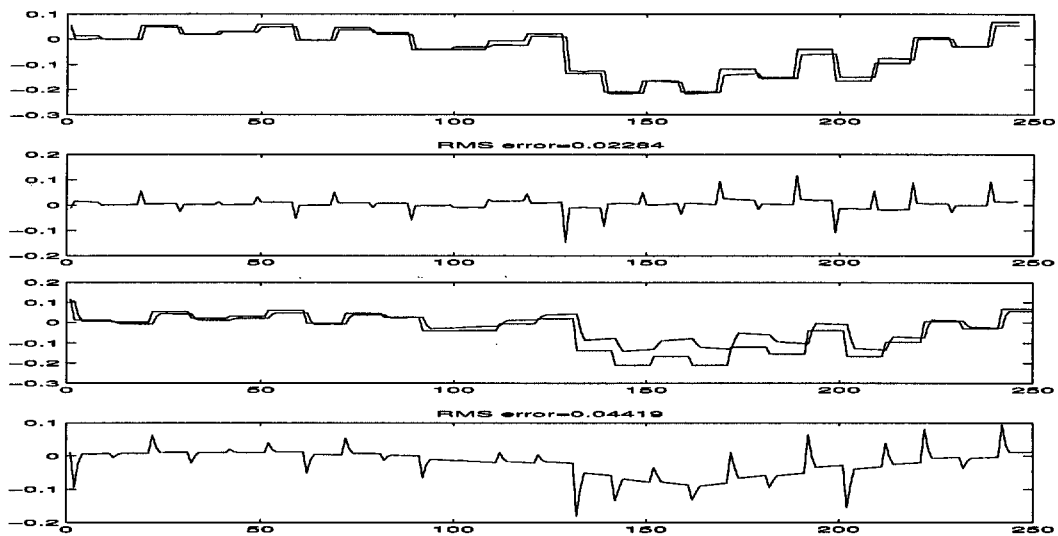


Fig. 5.12: Resultados de la identificación de la planta 2. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.

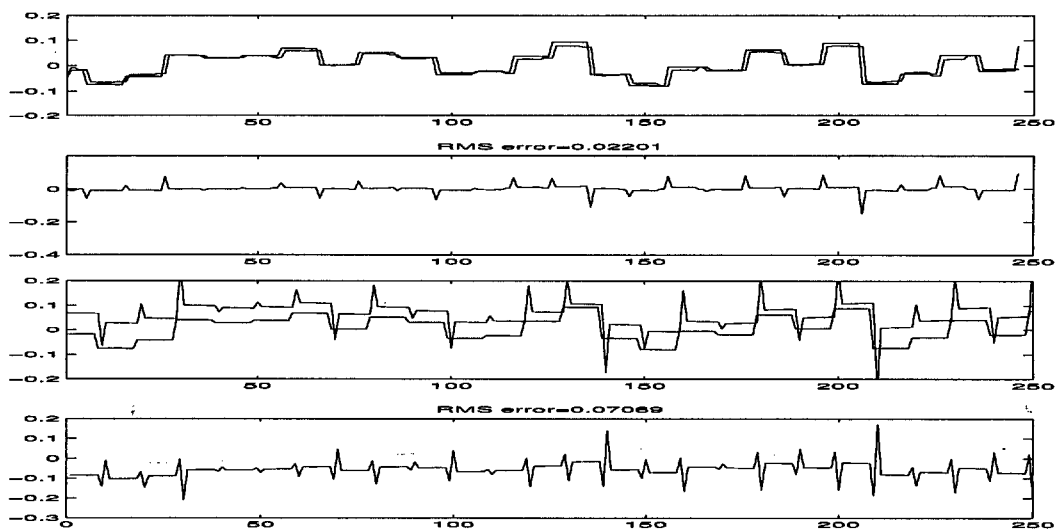


Fig. 5.13: Resultados de la identificación de la planta 3. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.

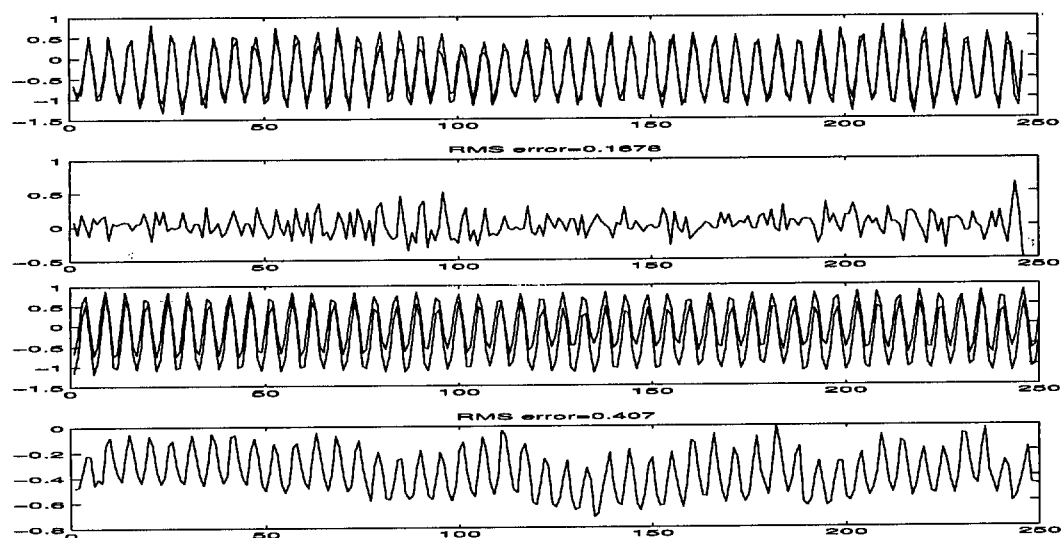


Fig. 5.14: Resultados de la identificación de la planta 4. Subfigura 1: salida del modelo vs. salida real para RGO. Subfigura 2: error cometido. Subfigura 3: salida del modelo vs. salida real para la Red Neuronal. Subfigura 4: error cometido.

5.8 Conclusiones

En este capítulo se ha desarrollado un nuevo algoritmo para la identificación de sistemas dinámicos basado en RGO. Se han utilizado modelos OE, ARMAX y NARMAX, pero el método se puede utilizar igualmente con cualquier otro método paramétrico.

En las simulaciones realizadas, los métodos basados en RGO mejoran considerablemente los resultados obtenidos por los métodos basados en mínimos cuadrados e incluso los de Redes Neuronales Backpropagation.

Los resultados obtenidos prueban que el método RGO puede ser utilizado para mejorar los resultados obtenidos con otros métodos de identificación de sistemas. En especial cuando el sistema no es lineal en los parámetros y cuando las señales de entrada y salida contienen ruido coloreado.

6. ESTIMACIÓN DE ESTADOS

6.1 Introducción

El filtro de Kalman da una solución efectiva al problema de filtrado lineal-Gaussiano. Sin embargo, cuando hay no linealidades, tanto sea en la especificación del modelo como en el proceso de observación, se requieren otros métodos.

Cuando el estado de un sistema tiene que ser estimado a partir de información sensorial con ruido, se necesita emplear algún tipo de estimador de estados para fusionar los datos de los diferentes sensores para producir una estimación precisa del verdadero estado del sistema.

El método mas frecuentemente utilizado para la fusión de la información dada por los sensores en aplicaciones de robótica móvil es el Filtro de Kalman [61] y sus diferentes variaciones. Este filtro se utiliza para combinar todos los datos de medidas (por ejemplo, para fundir los diferentes datos de los distintos tipos de sensores) para obtener una estimación óptima en sentido estadístico.

Si la dinámica del sistema y el modelo de observación son lineales y tanto el error del sistema como el error de medida (de los sensores) son Gaussianos, entonces el estimador de mínimos cuadrados puede ser calculado usando el filtro de Kalman. Es decir el Filtro de Kalman da una estimación óptima desde el punto de vista estadístico.

El cálculo del Filtro de Kalman se hace recursivamente, es decir, en cada iteración, sólo la nueva medida y la última estimación son usadas en el cálculo actual, por tanto no hay necesidad de guardar las estimaciones y medidas previas. Esta característica hace que no necesite ni una gran potencia de cálculo ni un gran almacenamiento.

Las medidas de un grupo de sensores pueden ser fundidas utilizando el filtro de Kalman para estimar el estado actual y para hacer una predicción del estado futuro del sistema.

En el caso no lineal el problema de filtrado en el espacio de estados viene dado por



$$x(k) = g(k, x(k-1), \varepsilon(k)) \quad (6.1)$$

$$z(k) = h(k, x(k), \eta(k)) \quad (6.2)$$

donde $\varepsilon(k)$ y $\eta(k)$ son los ruidos, que se suponen vectores aleatorios, independientemente distribuidos:

$$\begin{pmatrix} \varepsilon(k) \\ \eta(k) \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix} \right) \quad (6.3)$$

La estimación óptima implica la descripción de la densidad de probabilidad condicional

$$\hat{x}^{MMSE}(k|k) = E(x(k)|Z^k) = \int x(k)p(x(k)|Z^k)dx \quad (6.4)$$

Desafortunadamente, esta descripción necesita un número infinito de parámetros. Por esta razón, un número de aproximaciones subóptimas han sido propuestas. Estos métodos utilizan aproximaciones analíticas de las distribuciones de probabilidad, la ecuación de transición de estados o la ecuación de medida. Existen otros métodos, tales como el método de Montecarlo, que necesitan miles de puntos para aproximar la densidad de probabilidad condicional. En aplicaciones con muchas dimensiones, éstos últimos métodos no son prácticos. Por esta razón son necesarios métodos con un número razonable de operaciones, tales como los estudiados en este trabajo.

Para el problema de filtrado no-lineal, la aproximación más sencilla e intuitiva es utilizar el desarrollo en serie de Taylor y utilizar las funciones de medida y de transición no-lineales en el algoritmo del filtro lineal recursivo de Kalman. Los filtros no-lineales tradicionales incluyen el Filtro Extendido de Kalman (EKF), el Filtro No-lineal de Segundo orden (SNF) y el Filtro Iterado de una etapa (Single-stage Iteration Filter (SIF)), que están descritos en Wishner, Tabaczynski y Athans [106].

Para reducir el sesgo de las estimaciones del filtro y obtener el valor esperado de las distribuciones de probabilidad más correctamente, fue desarrollado el Filtro de Simulación Montecarlo por Tanizaki y Mariano [100, 101]

6.2 El Problema de Filtrado.

El problema consiste en hallar una estimación del estado $x(k)$ del sistema de interés, con una dinámica discreta no lineal dada por

$$x(k) = g(k, x(k-1), \varepsilon(k)) \quad (6.5)$$

donde g la función de estado del modelo, $x(k)$ es el estado del sistema en el tiempo k y $\varepsilon(k)$ es el ruido del proceso.

La única información disponible del sistema es la observación con ruido dada por la ecuación no lineal de medida

$$z(k) = h(k, x(k), \eta(k)) . \quad (6.6)$$

donde $z(k)$ es el vector de observación, h es el modelo de observación que transforma el espacio de estados en el espacio de observación y $\eta(k)$ es el ruido de medida. El estimador MMSE coincide con la media condicional. Sea $\hat{x}(i|j) \doteq E[x(i)|Z^j]$ con $Z^j = \{z(1), z(2), \dots, z(j)\}^T$. La covarianza estimada es $P_{xx}(i|j) = E[\{x(i) - \hat{x}(i|j)\}\{x(i) - \hat{x}(i|j)\}^T | Z^j]$. Estas ecuaciones son difíciles de evaluar en la práctica. Por esta razón se emplean estimadores recursivos. En el caso lineal con ruido aditivo gaussiano el MMSE es el filtro de Kalman. Las ecuaciones lineales son, en este caso:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)v(k), \quad (6.7)$$

$$P_{xx}(k|k) = P_{xx}(k|k-1) - W(k)P_{vv}(k|k-1)W^T(k) . \quad (6.8)$$

El vector $v(k)$ se llama innovación, y es la diferencia entre la observación y la predicción:

$$v(k) = z(k) - \hat{z}(k|k-1) . \quad (6.9)$$

La covarianza de esta magnitud es

$$P_{vv}(k|k-1) = P_{zz}(k|k-1) + R(k), \quad (6.10)$$

y $W(k)$ es la ganancia de Kalman y su valor es

$$W(k) = P_{xz}(k|k-1)P_{zz}^{-1}(k|k-1) . \quad (6.11)$$

6.2.1 La estimación lineal en sistemas dinámicos - El Filtro de Kalman

El problema de estimación dinámica

Consideremos un sistema dinámico lineal en tiempo discreto descrito por una ecuación vectorial en diferencias con ruido blanco Gaussiano aditivo que modela "perturbaciones no predecibles". La ecuación de estado es

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k), \quad k = 0, 1, \dots \quad (6.12)$$

donde $x(k)$ es el vector de estado de dimensión n_x , $u(k)$ es el vector de entrada de dimensión n_u (que es conocido), y $v(k)$, $k = 0, 1, \dots$, es la secuencia de ruido blanco Gaussiano de media cero con covarianza

$$E[v(k)v(k)'] = Q(k) \quad (6.13)$$

La ecuación de medida es

$$z(k) = H(k)x(k) + w(k) \quad k = 1, 2, \dots \quad (6.14)$$

con $w(k)$ secuencia de ruido de medida blanco Gaussiano de media cero con covarianza

$$E[w(k)w(k)'] = R(k) \quad (6.15)$$

Las matrices F , G , H , Q , y R se suponen conocidas y que pueden cambiar con el tiempo. En otras palabras, el sistema puede cambiar con el tiempo y los ruidos son no estacionarios.

El estado inicial $x(0)$ es en general desconocido y se modela como una variable aleatoria Gaussiana con media y covarianza conocidas. Las dos secuencias de ruido y el estado inicial se suponen mutuamente independientes y además los ruidos son independientes.

Las anteriores condiciones se llaman lineales-Gaussianas.

El algoritmo de estimación El algoritmo de estimación comienza con una estimación inicial $\hat{x}(0|0)$ de $x(0)$ y la covarianza inicial asociada $P(0|0)$, que se supone disponible.

Un ciclo del algoritmo de estimación dinámica (filtro de Kalman) consiste en estimar a partir

$$\hat{x}(k|k) = E[x(k)|Z^k] \quad (6.16)$$

que es la media condicional del estado en el tiempo k , dadas las observaciones hasta el tiempo k inclusive, y de la matriz de covarianza asociada

$$P(k|k) = E[(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))'|Z^k] \quad (6.17)$$

las correspondientes variables del siguiente paso, $\hat{x}(k+1|k+1)$ y $P(k+1|k+1)$.

Esto se puede hacer puesto que las variables Gaussianas quedan caracterizadas por sus dos primeros momentos.

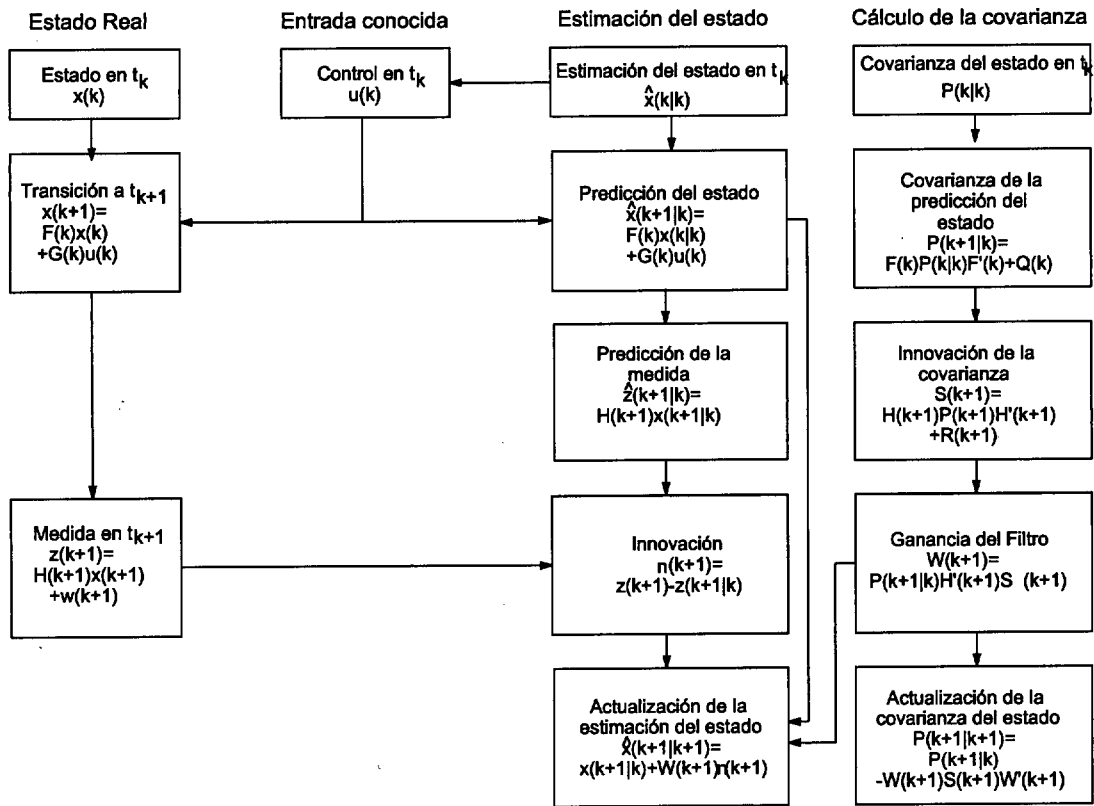


Fig. 6.1: Diagrama de flujo del filtro de Kalman.

Derivación del Algoritmo de Estimación Dinámica

La predicción del estado $\hat{x}(k+1|k) = E[x(k+1)|Z^k]$ se sigue de aplicar a la ecuación de estado $x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$ el operador de esperanza condicionada por Z^k ,

$$E[x(k+1)|Z^k] = E[F(k)x(k) + G(k)u(k) + v(k)|Z^k] \quad (6.18)$$

Como el ruido del proceso $v(k)$ es blanco y de media cero, entonces

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k) \quad (6.19)$$

Restando de la ecuación de estado obtenemos el error de predicción de estado,

$$\hat{x}(k+1|k) = x(k+1) - \hat{x}(k+1|k) = F(k)\hat{x}(k|k) + v(k) \quad (6.20)$$

La covarianza de la predicción de estado

$$P(k+1|k) = \text{cov}[x(k+1)|Z^k] = \text{cov}[\tilde{x}(k+1|k)|Z^k] \quad (6.21)$$

se obtiene de 6.20 ,

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \quad (6.22)$$

La covarianza entre el estado y la medida es , usando 6.20

$$\begin{aligned} E[\tilde{x}(k+1|k)\tilde{z}(k+1|k)'|Z^k] &= E[\tilde{x}(k+1|k)[H(k+1)\tilde{x}(k+1|k) + w(k+1)]'|Z^k] \\ &= P(k+1|k)H(k+1)' \end{aligned} \quad (6.23)$$

La ganancia del filtro, $P_{xz}P_{zz}^{-1} = W(k+1) = \text{cov}[x(k+1), z(k+1)|Z^k]S(k+1)^{-1}$ es

$$W(k+1) = P(k+1|k)H(k+1)'S(k+1)^{-1} \quad (6.24)$$

Por tanto, la estimación actualizada del estado puede ser escrita como

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1) \quad (6.25)$$

donde

$$v(k+1) = z(k+1) - \hat{z}(k+1|k) = \tilde{z}(k+1|k) \quad (6.26)$$

es la innovación. Finalmente, la covarianza actualizada es

$$\begin{aligned} P(k+1|k+1) &= P(k+1|k) - P(k+1|k)H(k+1)'S(k+1)^{-1}H(k+1)P(k+1|k) \\ &= [I - W(k+1)H(k+1)]P(k+1|k) \end{aligned} \quad (6.27)$$

o, en forma simétrica

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \quad (6.28)$$

6.3 Filtros No-lineales Tradicionales

6.3.1 Filtro Extendido de Kalman (EKF) y Filtro No-lineal de Segundo Orden (SNF)

El filtro Extendido de Kalman es similar a un filtro de Kalman linealizado, con la excepción de que la linealización es realizada en la trayectoria estimada en vez

de hacerlo en una trayectoria nominal previamente calculada. Por esta razón, las funciones $g(k, x(k-1), \varepsilon(k))$ y $h(k, x(k), \eta(k))$ son desarrolladas en serie de Taylor alrededor de $\hat{x}(k|k)$ con términos hasta primer o segundo orden para obtener el EKF o SNF respectivamente. El desarrollo con términos de segundo orden de la ecuación de transición es:

$$\begin{aligned}
 x(k) &= g(k, x(k-1), \varepsilon(k)) \simeq g(k, \hat{x}(k|k-1), 0) \\
 &+ g_x(k, x(k-1), \varepsilon(k))(x(k-1) - \hat{x}(k-1|k-1)) + g_\varepsilon(k, x(k-1), \varepsilon(k))\varepsilon(k) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' g_{xx}^j(k, x(k-1), \varepsilon(k)) (x(k-1) - \hat{x}(k-1|k-1)) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j \varepsilon'(k) g_{\varepsilon\varepsilon}^j(k, x(k-1), \varepsilon(k)) \varepsilon(k) \\
 &+ \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' g_{x\varepsilon}^j(k, x(k-1), \varepsilon(k)) \varepsilon(k) .
 \end{aligned} \tag{6.29}$$

y el desarrollo con términos hasta segundo orden de la ecuación de medida es:

$$\begin{aligned}
 z(k) &= h(k, x(k), \eta(k)) \simeq h(k, \hat{x}(k|k-1), 0) \\
 &+ h_x(k, x(k-1), \eta(k))(x(k-1) - \hat{x}(k-1|k-1)) + h_\eta(k, x(k-1), \eta(k))\eta(k) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' h_{xx}^j(k, x(k-1), \eta(k)) (x(k-1) - \hat{x}(k-1|k-1)) \\
 &+ \frac{1}{2} \sum_{j=1}^n e_j \eta'(k) h_{\eta\eta}^j(k, x(k-1), \eta(k)) \eta(k) \\
 &+ \sum_{j=1}^n e_j (x(k-1) - \hat{x}(k-1|k-1))' h_{x\eta}^j(k, x(k-1), \eta(k)) \eta(k) .
 \end{aligned} \tag{6.30}$$

donde e_j es el j^{th} vector de la base Cartesiana.

Las técnicas basadas en el Filtro de Kalman Extendido han demostrado ser robustas y mantener el seguimiento de la posición robot de forma precisa. Sin embargo el Filtro de Kalman Extendido no puede representar ambigüedades (como las simetrías) y no es capaz de relocalizar al robot en caso de fallo en la localización. Muchas de estas dificultades estriban en asumir la restricción Gaussiana del Filtro de Kalman.

6.3.2 Filtro extendido Iterado de Kalman (IEKF)

La predicción de la medida, hasta primer orden, es $\hat{z}(k|k-1) = h(k, \hat{x}(k|k-1))$. Hay errores en la predicción al usar $\hat{x}(k|k-1)$ para estimar $x(k)$. Otros errores adicionales son debidos a la no linealidad en la medida. Es posible disminuir estos errores si el estado actualizado no es calculado como una aproximación a la esperanza condicional sino como un estimador del tipo maximum a posteriori.

La función de densidad de probabilidad condicional, PDF, de $x(k)$ dado Z^k , si todas las variables aleatorias son Gaussianas, es

$$\begin{aligned} p(x(k)|Z^k) &= p(x(k)|z(k), Z^{k-1}) \\ &= \frac{1}{c} p(z(k)|x(k)) p(x(k)|Z^{k-1}) \\ &= \frac{1}{c} \mathcal{N}(z(k); h(k, x(k)), R(k)) \mathcal{N}(x(k); \hat{x}(k|k-1), P(k|k-1)) \end{aligned} \quad (6.31)$$

Maximizar esta función es equivalente a minimizar la siguiente

$$\begin{aligned} V(x(k)) &= \frac{1}{2} (z(k) - h(k, x(k)))' R(k)^{-1} (z(k) - h(k, x(k))) \\ &\quad + \frac{1}{2} (x(k) - \hat{x}(k|k-1))' (P(k|k-1))^{-1} (x(k) - \hat{x}(k|k-1)) \end{aligned} \quad (6.32)$$

El Filtro de Kalman Extendido Iterado (IEKF) usa un algoritmo Newton-Raphson para estimar $\hat{x}(k|k)$. Desarrollando V en serie de Taylor hasta segundo orden alrededor de la i -ésima estimación de $x(k)$ da como resultado:

$$V = V^i + V_x^i (x - x^i) + \frac{1}{2} (x - x^i)' V_{xx}^i (x - x^i) \quad (6.33)$$

Poniendo el gradiente a cero:

$$x^{i+1} = x^i - (V_{xx}^i)^{-1} V_x^i \quad (6.34)$$

Esto da lugar a la ecuación IEKF:

$$\begin{aligned} \hat{x}^i(k|k) &= \hat{x}^{i-1}(k|k) + P^{i-1}(k|k) H^{i-1}(k)' R(k)^{-1} \{z(k) - h(k, \hat{x}^{i-1}(k|k))\} \\ &\quad - P^{i-1}(k|k) P(k|k-1)^{-1} (\hat{x}^i(k|k) - \hat{x}(k|k-1)), \end{aligned} \quad (6.35)$$

with $H^{i-1}(k) = h_x(k, \hat{x}^i(k|k))$.

6.3.3 Filtro Iterado de una Etapa (Single-stage Iterated Filter (SIF))

Esta técnica trata de maximizar la densidad de probabilidad *a posteriori*. En el método EKF, la trayectoria nominal es linealizada alrededor de la trayectoria nominal determinada por $\hat{x}(k-1|k-1)$ y el estado nominal $\hat{x}(k|k-1)$. Si la trayectoria y el estado nominales no están cerca de la trayectoria verdadera, los desarrollos truncados del EKF presentarán pobres aproximaciones.

La ecuación $z(k) = h(k, x(k), \eta(k))$ puede ser reescrita como $q(k, z(k), x(k)) = \eta(k)$. Con esta expresión y $\hat{x}(k|k-1) = x(k) - \xi$ donde

$$\begin{pmatrix} \eta(k) \\ \xi(k) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} R(k) & 0 \\ 0 & P_{xz}(k|k-1) \end{pmatrix},$$

es posible minimizar la siguiente suma de cuadrados de los residuos con pesos:

$$q(k, z(k), x(k))' R^{-1}(k) q(k, z(k), x(k)) + (\hat{x}(k|k-1) - x(k))' P_{xz}(k|k-1) (\hat{x}(k|k-1) - x(k))$$

usando el método de optimización de Newton-Raphson.

$$\begin{aligned} \hat{x}^i(k|k-1) &= g^i(\hat{x}(k|k) + g_x^i(k, x(k|k-1), \epsilon(k))) (\hat{x}(k-1|k-1) - \hat{x}(k|k-1)), \\ P_{xx}^i(k|k-1) &= g_x^i(k, x(k|k), \epsilon(k)) P_{xx}^i(k|k-1) g_x^i(k, x(k|k-1), \epsilon(k))' + Q(k), \\ \hat{z}^i(k|k-1) &= h^i(k, x(k|k-1), \eta(k)), \\ P_{zz}^i(k) &= H u^i(k, \hat{x}(k|k), x) P_{xx}^i(k, k-1) (H u^i(k, x(k|k), x))' + R, \\ \tilde{z}^i(k) &= z(k) - \hat{z}^i(k|k-1), \\ W^i(k) &= P_{xx}^i(k, k-1) h_x^i(k, \hat{x}(k|k), x) P_{zz}^i(k), \\ P_{xx}^i(k, k) &= P_{xx}^i(k, k-1) - W^i(k) P_{zz}^i(k) W^i(k)', \\ x^i(k|k) &= x^i(k, k-1) + W^i(k) \tilde{z}^i(k). \end{aligned}$$

6.3.4 Filtro con Simulación Montecarlo (MSF)

Tanizaki y Mariano [101] propusieron un algoritmo en simulaciones estocásticas Montecarlo, donde las variables aleatorias normales están generadas por los términos de error $\epsilon(k)$ y $\eta(k)$ y las variables de estado $x(k)$ y $x(k-1)$ para evaluar los valores esperados mas exactamente.

Esta aproximación está basada en las ecuaciones:

$$\begin{aligned} x(k) &= g(k, \hat{x}(k|k-1), 0) \\ &+ g_x(k, x(k-1), \epsilon(k)) (x(k-1) - \hat{x}(k-1|k-1)), \end{aligned} \quad (6.36)$$



y

$$\begin{aligned} z(k) &= h(k, \hat{x}(k|k-1), 0) \\ &+ h_x(k, x(k-1), \eta(k))(x(k-1) - \hat{x}(k-1|k-1)) \end{aligned} \quad (6.37)$$

que son lineales en las variables de estado $x(k)$ y $x(k-1)$. Por tanto, este algoritmo es una combinación del Filtro Extendido de Kalman y la simulación estocástica Montecarlo. Este algoritmo es una combinación del EKF y de las simulaciones estocásticas Montecarlo.

Las aproximaciones del Filtro de Kalman se aproximan por caminos aleatorios:

$$\begin{aligned} \hat{x}(k|k-1) &= \frac{1}{n} \sum_{i=1}^n \hat{x}_i(k|k-1), \\ P_{xx}(k|k-1) &= \frac{1}{n} \sum_{i=1}^n (\hat{x}_i(k|k-1) - \hat{x}(k|k-1))(\hat{x}_i(k|k-1) - \hat{x}(k|k-1))', \\ \hat{z}(k|k-1) &= \frac{1}{n} \sum_{i=1}^n \hat{z}_i(k|k-1), \\ P_{zz}(k|k-1) &= \frac{1}{n} \sum_{i=1}^n (\hat{z}_i(k|k-1) - \hat{z}(k|k-1))(\hat{z}_i(k|k-1) - \hat{z}(k|k-1))', \\ P_{xz}(k|k-1) &= \frac{1}{n} \sum_{i=1}^n (\hat{z}_i(k|k-1) - \hat{z}(k|k-1))(\hat{x}_i(k|k-1) - \hat{x}(k|k-1))', \\ W(k) &= P_{xx}(k|k-1)P_{zz}(k|k-1), \\ P_{xx}(k|k) &= P_{xx}(k|k-1) - W(k)P_{zz}(k|k-1)W(k)', \end{aligned}$$

and

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)(\hat{z}_i(k|k-1) - \hat{z}(k|k-1)), \quad (6.38)$$

where $\hat{x}^i(k|k-1) = g(k, \hat{x}^i(k-1, k-1), \epsilon^i(k))$, and $\hat{z}^i(k, k-1) = h(k, \hat{x}^i(k, k-1), \eta^i(k))$.

6.4 El nuevo Filtro, RGO

Los algoritmos genéticos son un proceso probabilístico de búsqueda basado en la selección natural y en las leyes genéticas. La población $\mathcal{J} = (\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_n) \in J^N$ es modificada de acuerdo con los procesos evolutivos naturales: después de la inicialización, son ejecutados la selección $\omega : J^N \rightarrow J^N$, cruce $\chi : J^N \rightarrow J^N$, y mutación $\Xi : J^N \rightarrow J^N$ recursivamente en un bucle. Cada ejecución del bucle es llamada *generación* y \mathcal{J} representa la población en la generación τ .

El operador de selección tiene como fin mejorar la calidad media de la población dando a los individuos de mayor calidad una alta probabilidad de ser copiados en la nueva generación. La selección, por lo tanto, enfoca la búsqueda hacia regiones prometedoras del espacio de búsqueda. La calidad de un individuo es medida

por la función de salud $f : J \rightarrow \mathbb{R}$, donde J representa el espacio de todos los posibles individuos.

Los algoritmos genéticos se utilizan usualmente como método de optimización global de funciones invariantes en el tiempo, y usualmente se ejecutan 'off line'. Sin embargo, la selección natural de los seres vivos es un proceso local o semi local en el que las especies se adaptan por sí mismas al entorno, que a su vez es dependiente del tiempo (es 'on line').

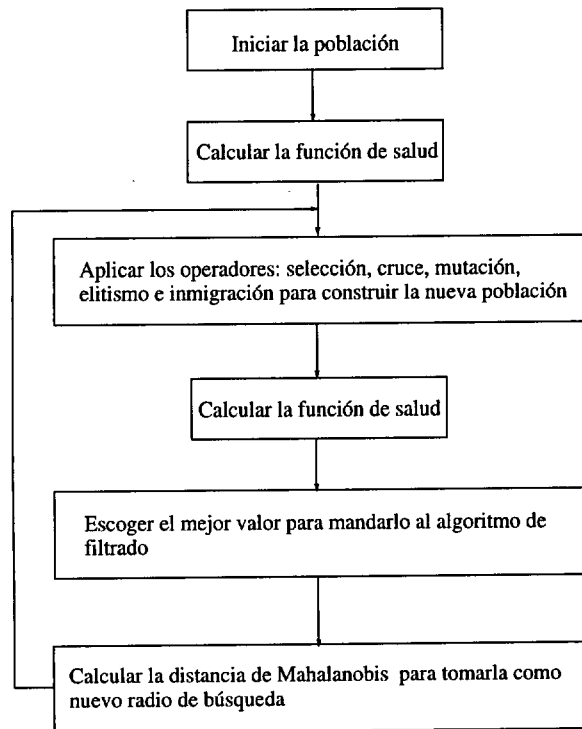


Fig. 6.2: Diagrama de flujo de la Optimización Genética Restringida aplicada a la estimación de estados.

Es posible adaptar el método de algoritmos genéticos si se restringe la búsqueda a un entorno de la estimación previa usando como función de salud:

$$f : B(\hat{x}(k|k-1), \sigma) \rightarrow \mathbb{R}$$

con $\sigma = \|P(k|k-1)\|$,

$$f(J) = \frac{1}{cte + V(\hat{x}(k|k-1))} \quad (6.39)$$

donde cte es una constante grande y

$$V(x(k)) = \frac{1}{2}(z(k) - h(k, x(k)))'R(k)^{-1}(z(k) - h(k, x(k))) + \frac{1}{2}(x(k) - \hat{x}(k|k-1))'(P(k|k-1)^{-1}(x(k) - \hat{x}(k|k-1))) \quad (6.40)$$

Por lo tanto el algoritmo actualiza las estimaciones de los estados y la matriz de covarianza y el bucle es repetido de nuevo, como se ilustra en la figura 6.2. La función de densidad de probabilidad de $x(k)$ dado Z^k viene dada como

$$\begin{aligned} p(x(k)|Z^k) &= p(x(k)|z(k), Z^{k-1}) \\ &= \frac{1}{c} p(z(k)|x(k)) p(x(k)|Z^{k-1}) \\ &= \frac{1}{c} N(z(k); h(k, x(k)), R(k)) N(x(k); \hat{x}(k|k-1), P(k|k-1)) \end{aligned} \quad (6.41)$$

Maximizar la función anterior es equivalente a calcular una estimación maximum a posteriori (MAP). Esto es también equivalente a minimizar $V(x(k))$, i.e. maximizar la función de salud $f(J)$.

La función de salud estándar (i.e. dividida por la suma de las saludes) es una aproximación de la función de densidad condicional (PDF).

$$p(x(k)|Z^k) = \frac{p(z(k)|x(k))p(x(k)|Z^{k-1})}{\int p(z(k)|x(k))p(x(k)|Z^{k-1})} \quad (6.42)$$

De todo lo anterior queda claro que es posible calcular precisamente las no linealidades de las funciones f y g , sin embargo la introducción de la hipótesis de ruido gaussiano no se puede evitar. Para determinar el radio de la zona de búsqueda, usamos la distancia de Mahalanobis

$$d = (\hat{x}(k|k-1) - \hat{x}(k-1|k-1))'P^{-1}(k|k)(\hat{x}(k|k-1) - \hat{x}(k-1|k-1)) \quad (6.43)$$

que mide la incertidumbre de la estimación $\hat{x}(k)$.

Todos estos procesos pueden ser aplicados al Filtro Extendido de Kalman (para obtener el Filtro de Optimización Genética Restringida, RGOF) o al Filtro No-lineal de Segundo Orden, SNF (para obtener el Filtro de Optimización Genética Restringida de Segundo Orden, SRGOF).

6.5 Comparación de Filtros No-lineales.

La comparación de filtros no-lineales la realizamos por medio de simulaciones Monte-Carlo. Un conjunto de datos $x(k)$ y $z(k)$ son simulados artificialmente y la

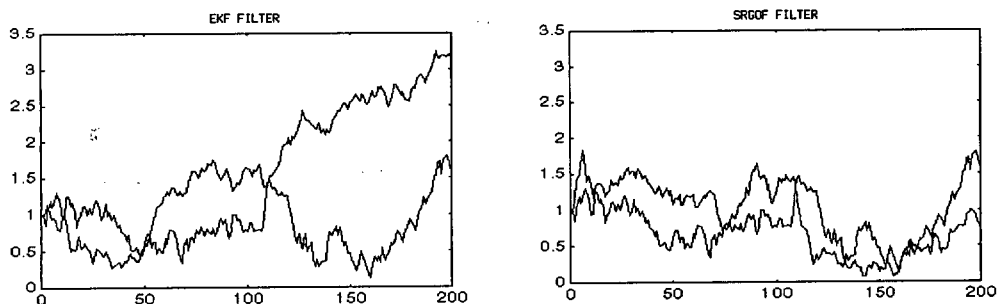


Fig. 6.3: Ejecuciones típicas para el modelo 5 con EKF y SRGOF para nivel de ruido 1.0.

estimación $\hat{x}(k)$ es comparada con $x(k)$, y el BIAS y el RMSE entre el valor estimado $\hat{x}(k)$ y el simulado $x(k)$ se calcula para cada tiempo k . Este procedimiento se realiza 6000 veces (30 ejecuciones de 200 puntos cada una).

Para comparar los distintos métodos de estimación, consideramos cinco ejemplos conocidos en orden ascendente de no linealidad y con tres niveles diferentes de ruido. En todos los casos $\varepsilon(k)$ y $\eta(k)$ se suponen normalmente distribuidos como

$$\begin{pmatrix} \varepsilon(k) \\ \eta(k) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}\right), \quad (6.44)$$

donde C es una constante y el valor inicial $x(0)$ está distribuido como una variable aleatoria normal.

El primer modelo es lineal:

$$\begin{cases} x(k) = x(k-1) + \varepsilon(k) \\ z(k) = x(k) + \eta(k) \end{cases} \quad (6.45)$$

El segundo es el modelo Logístico:

$$\begin{cases} x(k) = \frac{\exp(x(k-1))}{\exp(x(k-1)) + \exp(\varepsilon(k))} \\ z(k) = \frac{\exp(x(k))}{\exp(x(k)) + \exp(\eta(k))} \end{cases} \quad (6.46)$$

El tercero es el modelo ARCH:

$$\begin{cases} x(k) = (1 - b + b(x(k-1))^2)^{1/2} + \eta(k) \\ z(k) = x(k) + \varepsilon(k) \end{cases} \quad (6.47)$$

El cuarto es el modelo de Crecimiento No-estacionario de Kitagawa (1987) y Carlin et al. (1992).

$$\begin{cases} x(k) = 0.5x(k-1) + \frac{25x(k-1)}{1+x^2(k-1)} + 8\cos(1.2(t-1)) + \varepsilon(k) \\ z(k) = \frac{x^2(k)}{2} + \eta(k) \end{cases} \quad (6.48)$$

El último corresponde al seguimiento de un móvil con un sensor que mide solamente el ángulo, dado por Bar-Shalom y Fortmann [9] en 1988.

$$\begin{cases} x(k) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x(k-1) + \varepsilon(k) \\ z(k) = \text{atan}\left(\frac{20}{(x(1,k)-x_p(k))}\right) + \eta(k) \end{cases} \quad (6.49)$$

Tab. 6.1: Comparación de BIAS para el nivel de ruido 0.1.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	-0.2062	0.0058	-0.1275	0.7118	-0.2342
IEKF	-0.7654	0.0250	-1.2593	0.4796	-0.2488
SNF	-0.2063	0.0109	-0.1210	0.4796	-0.4361
SIF	0.1955	8.5E-05	-0.1812	0.5861	-0.3238
MSF	-4.9743	0.0057	-0.1397	0.4487	-0.2453
RGOF	-0.1889	0.0076	-0.1142	0.2688	0.2411
SRGOF	0.6469	0.0078	-0.1198	0.3893	0.0938

Tab. 6.2: Comparación de RMSE para el nivel de ruido 0.1.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	0.9616	0.0876	0.9162	10.9863	1.1963
IEKF	2.0934	0.0822	2.1868	10.8942	1.0293
SNF	0.9638	0.0536	0.8812	10.8942	0.9003
SIF	1.9693	0.0690	1.0947	13.1817	0.8980
MSF	36.5653	0.0786	0.8718	11.3058	0.8828
RGOF	0.9515	0.0532	0.8695	10.3893	1.1523
SRGOF	1.3145	0.0543	0.8554	10.1435	0.8511

Tab. 6.3: Comparación de BIAS para el nivel de ruido 0.5.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	-1.9802	0.2251	-0.1691	0.2727	0.1648
IEKF	-6.7135	0.2233	-0.7962	0.1858	-0.1300
SNF	-0.7908	0.2275	-0.1250	0.1858	-0.0797
SIF	0.4687	0.2269	-0.1911	0.8650	0.1989
MSF	-59.0465	0.2236	-0.0628	0.5375	-0.0202
RGOF	-0.8040	0.2209	-0.1453	0.4761	0.3438
SRGOF	-1.1171	0.2055	-0.1603	-1.2549	-1.0691

Tab. 6.4: Comparación de RMSE para el nivel de ruido 0.5.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	4.996	1.5537	0.8632	14.7955	5.5196
IEKF	10.068	1.5561	1.7991	14.7742	5.0794
SNF	4.354	1.5546	0.8463	14.7742	4.6296
SIF	8.767	1.5494	1.0864	17.9635	4.4094
MSF	154.528	1.5540	0.8523	14.8041	5.1212
RGOF	3.847	1.5525	0.8573	10.3697	5.9636
SRGOF	4.2704	1.5063	0.8611	6.0052	4.8815

Tab. 6.5: Comparación de BIAS para el nivel de ruido 1.0.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	-1.0107	0.0811	-0.1275	0.28454	0.2183
IEKF	-4.0117	0.1628	-1.2593	0.19430	0.2765
SNF	-1.0325	0.0859	-0.1210	0.19430	0.0084
SIF	0.9615	0.0369	-0.1812	0.52118	-1.6102
MSF	-17.5458	0.0810	-0.1397	0.43729	0.6963
RGOF	-2.3808	0.0524	-0.1146	0.01695	0.3160
SRGOF	-2.1398	0.0529	-0.1198	0.11951	1.7845

6.6 Discusión

En este capítulo se describe un nuevo método de filtrado no lineal. Este método utiliza la Optimización Genética Restringida para reducir el error de estimación

Tab. 6.6: Comparación de RMSE para el nivel de ruido 1.0.

	Model 1	Model 2	Model 3	Model 4	Model 5
EKF	4.8071	0.2457	0.9162	14.5453	11.0601
IEKF	9.7702	0.2791	2.1868	14.5721	9.7927
SNF	4.8194	0.2374	0.8812	14.5721	8.2852
SIF	9.8352	0.3276	1.0947	15.6697	8.2340
MSF	173.3967	0.2417	0.8718	14.6868	8.1326
RGOF	5.5300	0.2286	0.8696	10.1305	10.2766
SRGOF	5.9861	0.2287	0.8554	9.9886	9.1265

de las estimaciones previas EKF o SNF.

Las tablas 1-3 resumen los resultados de cinco modelos con tres niveles de ruido: $\|\epsilon\| = \|\eta\| = 0.1, 0.5$ y 1.0 . Cada número mostrado en las tablas representa la media de treinta ejecuciones de doscientos puntos cada una.

Queda claro de los siguientes resultados que la precisión (basada en el criterio RMSE) de las estimaciones del SRGOF son ciertamente superiores a las de los otros algoritmos, especialmente en las situaciones con más ruido y más no linealidades. Juzgando a partir del criterio de los BIAS no hay grandes diferencias entre los diferentes métodos, aunque algunos algoritmos (MSF, SIF y IEKF) son menos robustos en ciertas situaciones.

Finalmente, la característica más importante del método propuesto SRGOF es su robustez en las situaciones de alto ruido y fuertes no linealidades, que son las características deseadas de tales filtros.

7. IDENTIFICACIÓN EN EL ESPACIO DE ESTADOS

7.1 Modelos de Sistemas Lineales en el Espacio de Estados

Un sistema lineal general discreto e invariante en el tiempo puede ser descrito como

$$y(t) = G_0(q)u(t) + H_0(q)e_0(t), \quad (7.1)$$

donde $y(t)$ es el vector de salida de dimensión p , $u(t)$ es el vector de entrada de dimensión m y que se suponen conocidos. Las perturbaciones desconocidas que actúan en la salida se suponen generadas por el segundo término, donde $e_0(t)$ es el vector de ruido de dimensión p . Se supone que $e_0(t)$ es una secuencia de variables estocásticas independientes que cumplen

$$E[e_0(t)] = 0, \quad \text{y} \quad E[e_0(t)e_0(t)^T] = \Lambda_0 \quad (7.2)$$

donde E es el operador esperanza matemática.

Como el sistema es de orden finito, puede ser descrito también en el espacio de estados, introduciendo el vector auxiliar de estados $x(t)$ de dimensión n , siendo n la dimensión del sistema. La relación de entrada-salida puede ser descrita por

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ke_0(t) \\ y(t) &= Cx(t) + e_0(t). \end{aligned} \quad (7.3)$$

donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $K \in \mathbb{R}^{n \times p}$ y $C \in \mathbb{R}^{p \times n}$.

La manera de representar el ruido en 7.3 se dice que está en forma de innovaciones. El modelo en espacio de estados 7.3 es igual al sistema 7.1 si las matrices A , B , C y K cumplen

$$G_0(q) = C(qI - A)^{-1}B, \quad \text{y} \quad H_0(q) = C(qI - A)^{-1}K + I. \quad (7.4)$$

Las matrices A , B , C y K no son únicas. Es posible un cambio de base para los estados

$$x'(t) = T^{-1}x(t) \quad (7.5)$$

donde $T \in \mathbb{R}^{n \times n}$ es una matriz no singular, y el sistema en las nuevas variables de estado son

$$\begin{aligned} x'(t+1) &= A'x'(t) + B'u(t) + K'e_0(t) \\ y(t) &= C'x'(t) + e_0(t). \end{aligned} \quad (7.6)$$

Donde $A' = T^{-1}AT$, $B' = T^{-1}B$, $K' = T^{-1}K$ y $C' = CT$.
Es fácil ver que

$$\begin{aligned} C(qI - A)^{-1}B &= C'(qI - A')^{-1}B' \\ C(qI - A)^{-1}K + I &= C'(qI - A')^{-1}K' + I \end{aligned} \quad (7.7)$$

7.2 Parametrización e Identificabilidad

Para la identificación de sistemas es deseable frecuentemente utilizar modelos paramétricos, es decir, modelos que pueden ser descritos por un cierto número de parámetros con valores reales, que pueden ser considerados como un vector de parámetros $\theta \in \mathbb{R}^d$. Un modelo en particular viene descrito por un valor del vector d -dimensional θ . La correspondencia del espacio de parámetros \mathbb{R}^d al espacio de modelos lineales se le llama *estructura de modelos* \mathcal{M} .

Consideramos, por tanto, las *estructuras de modelos* en el espacio de estados y en la forma de innovaciones:

$$\mathcal{M} : \begin{cases} \hat{x}(t+1) = A(\theta)\hat{x}(t) + B(\theta)u(t) + K(\theta)e(t) \\ y(t) = C(\theta)\hat{x}(t) + e(t). \end{cases} \quad (7.8)$$

donde las matrices A, B, C y K están construidas a partir del vector θ de acuerdo con la estructura de modelos \mathcal{M} .

Una estructura de modelos \mathcal{M} es globalmente identificable si $\mathcal{M}(\theta) = \mathcal{M}(\theta^*) \Rightarrow \theta = \theta^*$. Es decir, dos vectores de parámetros distintos θ y θ^* no pueden dar el mismo modelo.

Las estructuras de modelos identificables han sido muy utilizadas para la identificación de sistemas debido a la correspondencia biunívoca entre los modelos y los valores del vector de parámetros. Sin embargo, para algunos sistemas es difícil encontrar una parametrización bien condicionada.

Las estructuras de modelo identificables \mathcal{M}_I fueron introducidas en [88] y pueden ser definidas como:

Definición 7.2.1: Sea $A(\theta)$ una matriz rellena de ceros y con unos en la línea superior a la diagonal. Sean las filas numeradas r_1, r_2, \dots, r_p , donde $r_p = n$, rellenas con parámetros. Sean $B(\theta)$ y $K(\theta)$ rellenas con parámetros y $C(\theta)$ con ceros y con unos en la fila i , columna $r_{i-1} + 1$. (Se considera $r_0 = 0$ y que p es la dimensión de $y(t)$).

Ejemplo Una parametrización identificable en variables de estado de un sistema con cuatro estados, una entrada y una salida con índices 1,4 está dada por

$$\hat{x}(t+1) = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \theta_5 & \theta_6 & \theta_7 & \theta_8 \end{pmatrix} \hat{x}(t) + \begin{pmatrix} \theta_9 \\ \theta_{10} \\ \theta_{11} \\ \theta_{12} \end{pmatrix} u(t) + \begin{pmatrix} \theta_{13} & \theta_{14} \\ \theta_{15} & \theta_{16} \\ \theta_{17} & \theta_{18} \\ \theta_{19} & \theta_{20} \end{pmatrix} e(t) \quad (7.9)$$

$$\hat{y}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \hat{x}(t) + e(t). \quad (7.10)$$

Ljung [69] demuestra el siguiente teorema, 4A.2.

Teorema 7.2.1: La estructura de modelos en el espacio de estados \mathcal{M}_I anteriormente definida es globalmente identificable en θ^* si y sólo si $\{A(\theta^*), [B(\theta^*)K(\theta^*)]\}$ es controlable.

7.3 Modelos en el Espacio de Estados Totalmente Parametrizados

Si consideramos el modelo 7.3 con todas las matrices A , B , C y K rellenas con parámetros, estamos sobre-parametrizando el sistema y perdiendo identificabilidad. Más exactamente, para la estructura de modelos identificable se necesitan $2np + nm$ parámetros y el número de parámetros para la estructura de modelos totalmente parametrizada es

$$d = n^2 + 2np + nm. \quad (7.11)$$

Es decir, para la estructura de modelos totalmente parametrizada se necesitan n^2 parámetros más que para la estructura de modelos identificable.

No obstante, esta estructura de modelos en el espacio de estados tiene las ventajas de que no tenemos que buscar cuál de todas las diferentes formas de los



multi-índices se ajusta mejor a nuestro sistema puesto que la estructura totalmente parametrizada contiene a todas las demás. Además la calidad del modelo puede aumentar al elegir una estructura más flexible. Esto nos permite además hacer transformaciones para obtener una descripción que esté bien condicionada numéricamente.

Basándonos en las *realizaciones balanceadas*, introducidas por Moore [82], Kabamba [59] describe una parametrización canónica identificable para sistemas en tiempo continuo con valores singulares de Hankel distintos. Este resultado fue generalizado por Ober a sistemas arbitrarios en [86] incluyendo una extensión a sistemas en tiempo discreto utilizando la transformación bilineal. con ésta parametrización los parámetros pueden ser variados de forma continua sin que el sistema resulte no mínima. Otra ventaja, en comparación con la forma observable, es que las parametrizaciones balanceadas tienen mejores propiedades numéricas. Sin embargo, los índices estructurales deben ser conocidos también para este tipo de parametrización. Chou [17] usa la parametrización equilibrada para la identificación de sistemas.

7.4 Realizaciones Equilibradas (Balanced Realizations)

En las realizaciones no equilibradas, pequeños cambios en un parámetro pueden dar grandes cambios en el comportamiento del sistema. Esto debe ser evitado en cualquier método de identificación.

Un ejemplo de realización no equilibrada es el siguiente. Consideremos el sistema $G(q) = \frac{1}{(q-0.99)^4}$ que en forma de estructura identificable tiene una matriz A igual a:

$$A(\theta) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.9606 & 3.8812 & -5.8806 & 3.9600 \end{pmatrix} \quad (7.12)$$

Los valores propios, que son todos ellos iguales a 0.99, son extremadamente sensible a perturbaciones en los parámetros. Un cambio aditivo de 2×10^{-8} en cualquiera de los parámetros puede perturbar a uno de los autovalores y convertirlo en mayor que uno. Por esta razón es necesario introducir el concepto de realización equilibrada.

La matriz

$$W - 0 = \sum_{k=0}^{\infty} (A^T)^k C^T C A^k$$

es conocida como el *Gramiano de observabilidad* para el sistema en el espacio de estados. Los autovalores de esta matriz describen cómo la variable de estado inicial $x(0)$ influencia la señal de salida $y(t)$ cuando $u(t) \equiv 0$. Esta matriz también satisface la siguiente ecuación de Lyapunov

$$W_0 = A^T W_0 A + C^T C.$$

La matriz dual

$$W_c = \sum_{k=0}^{\infty} A^k B B^T (A^T)^k$$

es llamada el *Gramiano de controlabilidad*. Esta matriz describe cómo la entrada u influencia al vector de estado x . W_c también satisface

$$W_c = A W_c A^T + B B^T.$$

Las matrices Gramianas son simétricas por construcción.

Definición. Una realización en el espacio de estados es equilibrada si

$$W_o = W_c = \Sigma = \text{diag} (\sigma_1 \dots, \sigma_n)$$

con $\sigma_1 \geq \dots \geq \sigma_n > 0$.

7.5 Identificación RGO en el Espacio de Estados

La entrada excitadora $u(t)$ se ha escogido como una secuencia binaria pseudo-aleatoria (PRBNS), con ruido coloreado.

Como primer paso, para hacer esta identificación paramétrica, realizamos una primera identificación de múltiples modelos de tipo ARX, $A(q^{-1})y_k = z^{-d}B(q^{-1})u_k + e(t)$, mediante el método de mínimos cuadrados, y después se transforma en una realización en el espacio de estados.

La estimación es transformada en una realización balanceada para tener un modelo mejor condicionado numéricamente.

A continuación realizamos la búsqueda de modelos que se vayan adaptando mejor al sistema por medio de un algoritmo genético.

Para hacer la función de pérdida $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_{n-k})^2$ lo más pequeña posible, se ha utilizado como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde "cte" es

una constante suficientemente grande (10^8) para mantener la diversidad de la población. La probabilidad de reproducción se ha calculado como $F_n(t) = F(t)/\bar{F}$.

La búsqueda es llevada a cabo usando matrices completamente parametrizadas y a cada generación la estimación es transformada en una realización balanceada para conseguir un modelo bien condicionado numéricamente.

En cada generación, sustituimos el anterior modelo por el mejor de la nueva generación como centro del entorno donde se realiza la búsqueda, de esta manera, en caso de un sistema LTI, el mejor modelo debe ir tendiendo al valor que minimiza el funcional y la población iría tendiendo a cero. A continuación realizamos una nueva lectura de la entrada y la salida del sistema.

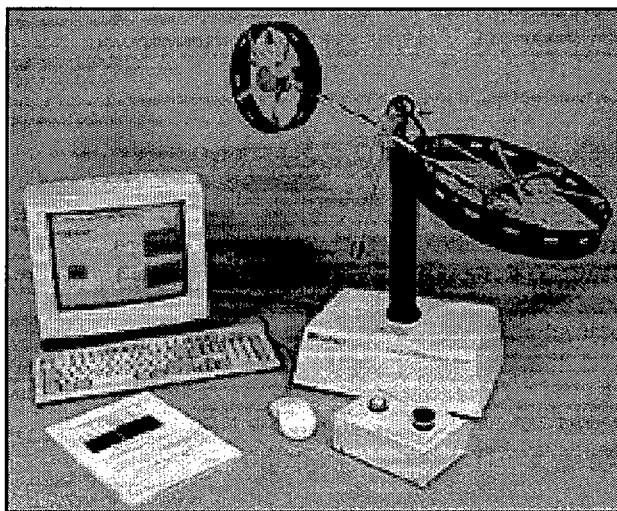


Fig. 7.1: *Twin Rotor MIMO System (TRMS).*

De esta manera conseguimos un conjunto de aproximaciones sucesivas a mejores modelos en caso de el sistema sea invariante en el tiempo y si el sistema es variable, pero de variación lo suficientemente lenta, entonces el modelo va siguiendo al sistema en sus variaciones.

Podemos además aprovechar las características convenientes de la convergencia global si al principio tomamos una bola lo suficientemente grande, de manera que reduzcamos la probabilidad de quedarnos en un mínimo local. Una vez que sepamos que seguimos al mínimo adecuado, podemos reducir el tamaño de la bola y al mismo tiempo de la población, para conseguir una mayor velocidad de ejecución del algoritmo.

Para que el algoritmo no tenga una convergencia prematura y se homogeneice demasiado la población, podemos aumentar el parámetro de probabilidad de mutación.

Aunque este algoritmo es costoso en cuanto a operaciones, es lo suficientemente rápido como para ser empleado en línea. A esto hay que añadir que las aplicaciones a los que típicamente se desea aplicar este identificador, es decir a los sistemas de control adaptativo, poseen dos bucles, uno rápido, que es el de control, y otro lento que es el identificador. Por tanto se puede aplicar a sistemas con una constante de tiempo del orden de las centésimas de segundo.

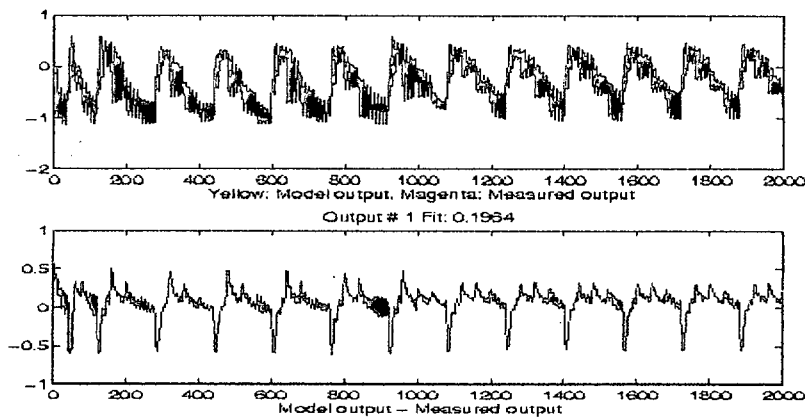


Fig. 7.2: Resultados de identificación de la Planta con el método ARX. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error

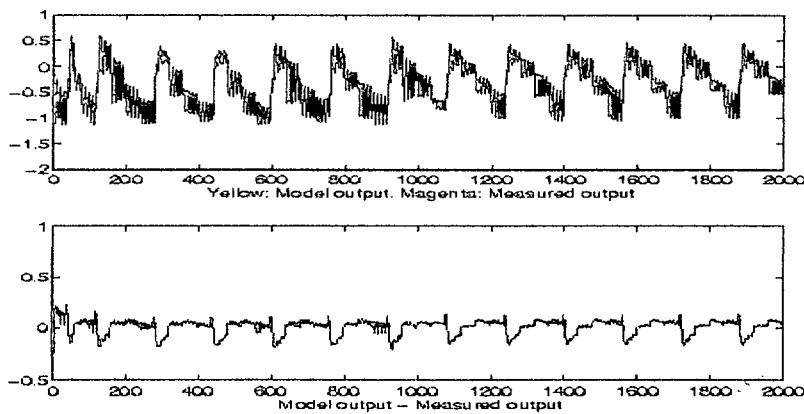


Fig. 7.3: Resultados de identificación de la Planta con el método Gauss-Newton. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error

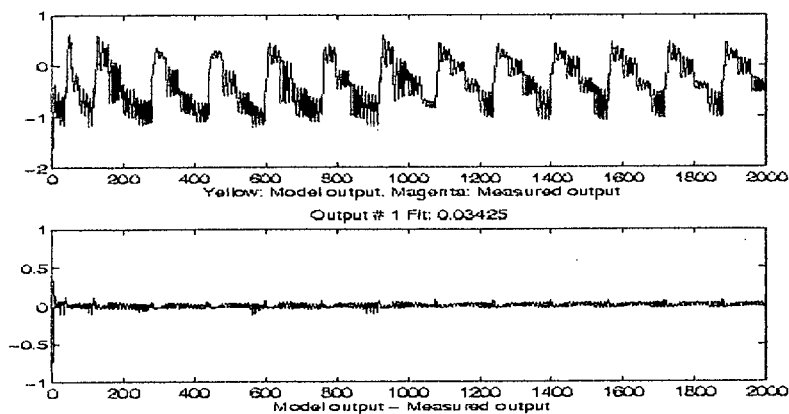


Fig. 7.4: Resultados de identificación de la Planta con el método RGO. Arriba: Salida del Modelo vs. Salida del Sistema. Abajo: Error

7.5.1 El Algoritmo

1. Se obtiene una primera estimación utilizando mínimos cuadrados de un modelo ARX y se pasa al espacio de estados.
2. Convertimos el sistema en una realización balanceada.
3. Hacemos $k = 1$
4. Resolvemos el problema de minimización

$$\hat{\theta}^k = \arg \min_{\theta} V_N(\theta) \quad (7.13)$$

empleando RGO (Restricted Genetic Optimization) con matrices completamente parametrizadas.

5. Convertimos la estimación obtenida en una realización balanceada.
6. Hacemos $k = k + 1$ y repetimos los pasos 4-6 hasta que

$$V_N(\hat{\theta}_b^{k-2}) - V_N(\hat{\theta}_b^{k-1}) < \varepsilon \quad (7.14)$$

donde ε es una constante dada a priori.

7.6 Resultados

Para poder contrastar los resultados con otros métodos, vamos a utilizar como índices de funcionamiento los siguientes: Error RMS:

$$\text{RMS error} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (7.15)$$

Error relativo RMS:

$$\text{Relative RMS error} = \frac{\sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{N}}}{\sqrt{\frac{\sum_{i=1}^n (y_i)^2}{N}}} \times 100\% \quad (7.16)$$

Para comparar los varios métodos de identificación, consideramos la aplicación a sistemas reales: al Twin Rotor MIMO System (TRMS) que es un sistema dinámico de laboratorio diseñado especialmente para experimentos de control.

En ciertos aspectos su comportamiento es parecido al de un helicóptero. Desde el punto de vista de control ejemplifica un sistema no-lineal de orden alto con diferente dinámica en cada posible posición.

Los resultados obtenidos de los Errores Cuadráticos Relativos Medios (RMSE) se pueden resumir en la siguiente tabla:

Error RMS relativo	
ARX331	0.1964
ARX441	0.1723
N4SID6	0.1695
Gauss-Newton331	0.06425
RGO331	0.03378

Tab. 7.1: Errores Cuadráticos Relativos

7.7 Discusión

En este capítulo se han comprobado las posibilidades que tiene la estructura de modelo totalmente parametrizada y su identificación por medio de los métodos ARX, Gauss-Newton y RGO. La principal idea es que esta estructura de modelo cubre todos los sistemas de un orden dado.

El método de identificación que se presenta utiliza la realización equilibrada (balanced realization) para conseguir un modelo bien condicionado numéricamente.

En la tabla de resultados se puede observar que el método RGO se comporta mejor que los otros algoritmos en un sistema como el TRMS con no linealidades y ruido.

8. APLICACIONES AL CONTROL PREDICTIVO

8.1 Introducción al Control Predictivo Basado en Modelo

Bajo este nombre se agrupan una serie de estrategias de control que tienen en común como característica principal, el hacer uso explícito de un modelo del proceso, para obtener la señal de control mediante la minimización de una cierta función de coste. En este sentido forman parte de una familia más amplia, en la que se hace uso de un modelo del proceso para diseñar el controlador: LQ, IMC, asignación de polos y ceros, etc.

Además de garantizar la operación estable de la planta, los controladores avanzados de hoy en día han de satisfacer una serie de criterios adicionales u objetivos secundarios de control: económicos, de seguridad, limitaciones físicas de los equipos, calidad del producto final, regulaciones ambientales, preferencias humanas, etc.

Muchos de estos modelos admiten una representación matemática muy natural, bajo la forma de funciones objetivo dinámicas y ligaduras dinámicas de tipo desigualdad.

El control predictivo basado en modelo tiene las siguientes características:

- Uso explícito de un modelo para predecir las salidas futuras.
- Cálculo de cierta secuencia que minimice cierta función objetivo.
- El horizonte se va desplazando hacia el futuro, lo que implica la aplicación de la primera señal de control calculada en cada paso.

Ventajas del control predictivo basado en modelos sobre otros métodos:

- Se puede aplicar con pocos conocimientos de control, porque los conceptos son intuitivos y el sintonizado es relativamente sencillo.

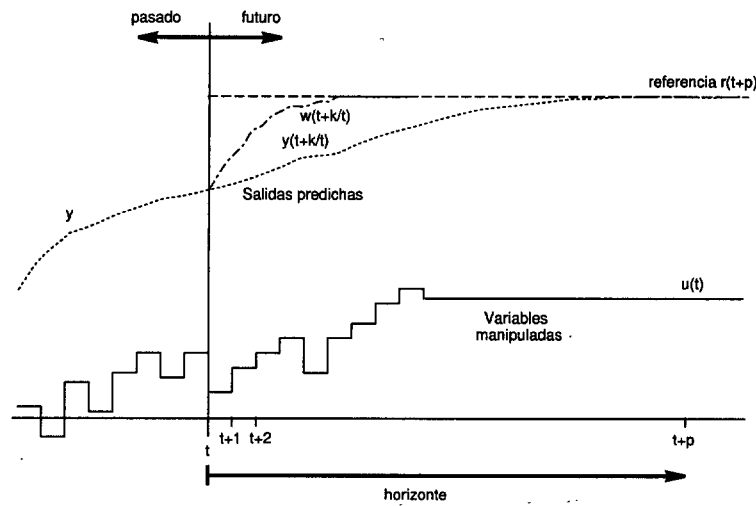


Fig. 8.1: La metodología del control predictivo basado en modelos.

- Se puede utilizar para controlar una gran cantidad de procesos, tanto sencillos como complejos; incluyendo sistemas con tiempo de retardo grandes y sistemas de fase no mínima.
- Se puede aplicar al caso multivariable.
- El controlador resultante es una sencilla ley de control lineal.
- Su extensión para tratar el caso con restricciones es conceptualmente sencilla y puede ser incluida durante el diseño.
- Es muy útil cuando las referencias futuras son conocidas.
- Es una metodología abierta.

8.1.1 Estrategias de Control Predictivo Basado en Modelo

1. En cada instante de tiempo t , se predicen las salidas futuras del proceso, $y(t+kl)$ para una cierta ventana de tiempo $k=1,2,\dots,N$, utilizando el modelo del proceso. N recibe el nombre de horizonte de predicción. Estas salidas se predicen con la información que está disponible en el instante t , y con los valores futuros de un escenario de control que se postula en el instante t .

8.1. INTRODUCCIÓN AL CONTROL PREDICTIVO BASADO EN MODELO 109

2. Se define una trayectoria de referencia, $r(t+kl)$, que describe cómo se desea guiarla salida del proceso desde su valor actual, $y(t)$ hacia sus puntos de consigna futuros, $w(t+kl)$.
3. De todo este conjunto se deduce la secuencia de control $u(t+kl)$ que hace mínima la función objetivo.
4. La señal de control $u(t/t)$ se manda al proceso. El resto de las señales se rechazan ya que $y(t+1)$ es conocido y se vuelve al punto 1.

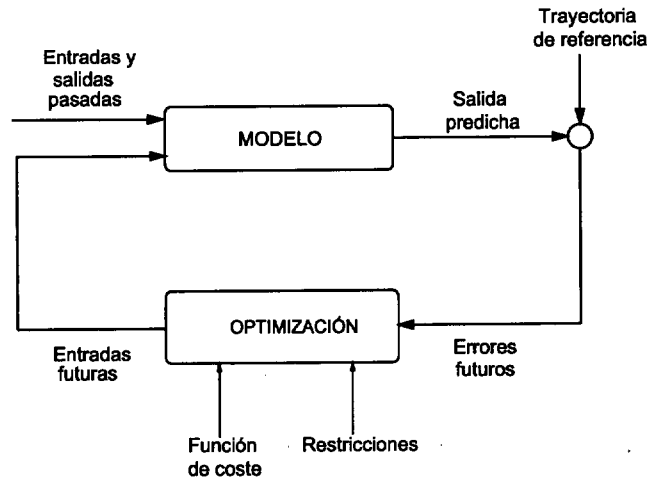


Fig. 8.2: Estructura básica del control predictivo.

8.1.2 La función objetivo

El criterio más sencillo para el control predictivo es :

$$J = \sum_{i=1}^{H_p} [\hat{y}(t+1) - \omega(t+i)]^2 \quad (8.1)$$

Con este modelo se obtiene el control predictivo de mínima varianza.

La expresión que se utiliza con más frecuencia es :

$$J(N_1, N_2, N_3) = E \left\{ \sum_{j=N_1}^{N_2} \delta(j) [y(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad (8.2)$$

N_1 y N_2 : Horizonte de coste (mínimo y máximo).

N_u : Horizonte de control.



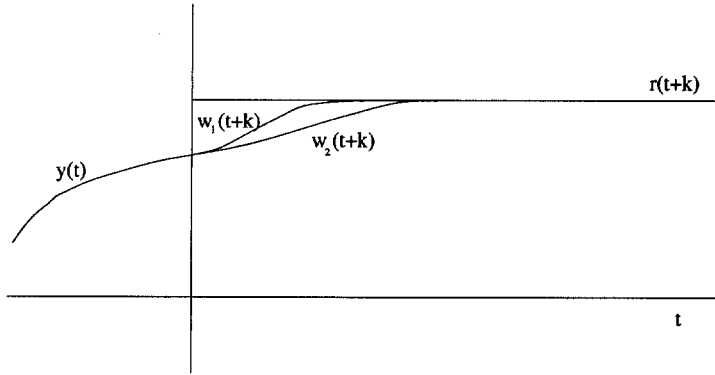


Fig. 8.3: Distintas trayectorias de referencia.

8.1.3 Trayectoria de referencia

Si conocemos la evolución futura de la referencia, podemos actuar antes de que los cambios se produzcan.

8.2 Control Predictivo GPC.

Partimos de un modelo ARIMAX (también llamado CARIMA):

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{\varepsilon(k)}{\Delta} \quad (8.3)$$

en donde la perturbación no medible viene representada por una secuencia de ruido blanco de media cero $\xi(t)$ coloreada por el polinomio $C(z^{-1})$ e integrada por el operador incremento δ .

Para optimizar la función de coste:

$$J(N_1, N_2, N_3) = E\left\{ \sum_{j=N_1}^{N_2} \delta(j)[y(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \right\} \quad (8.4)$$

consideramos la ecuación diofantina:[2]

$$1 = E_j(z^{-1})\Delta A(z^{-1}) + z^{-j}F_j(z^{-1}); \quad 1 = E_j(z^{-1})\tilde{A}(z^{-1}) + F_j(z^{-1}) \quad (8.5)$$

Si multiplicamos [1] por $\Delta E_j(z^{-1})z^{-j}$ y consideramos [2]

$$\tilde{A}(z^{-1})E_j(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) + E_j(z^{-1})e(t+j) \quad (8.6)$$

$$(1 - z^{-j}F_j(z^{-1}))y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) + E_j(z^{-1})e(t+j) \quad (8.7)$$

$$y(t+j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) + E_j(z^{-1})e(t+j) \quad (8.8)$$

Como el grado de E_j es $j-1$, los términos de ruido están todos en el futuro. La mejor predicción es:

$$\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})y(t) \quad (8.9)$$

con

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1}) \quad (8.10)$$

8.3 Ejemplo.

Considérese el siguiente sistema, descrito mediante un modelo ARIMAX:

$$y(t) = 1.7y(t-1) - 0.7y(t-2) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + \xi(t) \quad (8.11)$$

con $\xi(t)$ una señal de ruido blanco de media cero. En la notación empleada habitualmente, los polinomios de la función de transferencia de este sistema son:

$$\begin{cases} A(q^{-1}) = 1 - 1.7q^{-1} + 0.7q^{-2} \\ B(q^{-1}) = 0.9q^{-1} - 0.6q^{-2} \end{cases} \quad (8.12)$$

y se ha tomado $C(q^{-1}) = 1$. Los parámetros del controlador GPC son $N_1 = 1$, $N_u = N_2 = 3$ y $\lambda = 0.1$.

Las ecuaciones del predictor se pueden obtener resolviendo la ecuación diofantina asociada, pero para resolverla 'a mano', es más fácil iterar. Así,

$$\begin{cases} y(t+1|t) = 1.7y(t) - 0.7y(t-1) + 0.9\Delta u(t|t) - 0.6\Delta u(t-1) \\ y(t+2|t) = 1.7y(t+1|t) - 0.7y(t) + 0.9\Delta u(t+1|t) - 0.6\Delta u(t|t) \\ \quad = 2.19y(t) - 1.19y(t-1) + 0.9\Delta u(t+1|t) + 0.93\Delta u(t|t) \\ \quad \quad - 1.02\Delta u(t-1) \\ y(t+2|t) = 1.7y(t+2|t) - 0.7y(t+1|t) + 0.9\Delta u(t+2|t) - 0.6\Delta u(t-1|t) \\ \quad = 2.533y(t) - 1.533y(t-1) + 0.9\Delta u(t+2|t) + 0.93\Delta u(t+1|t) \\ \quad \quad + 0.951\Delta u(t|t) - 1.314\Delta u(t-1) \end{cases} \quad (8.13)$$

en donde se ha tomado $x_i(t+j|t) = 0$, Agrupando coeficientes, resulta

$$\begin{pmatrix} y(t+1|t) \\ y(t+2|t) \\ y(t+3|t) \end{pmatrix} = \begin{pmatrix} 0.9 & 0 & 0 \\ 0.93 & 0.9 & 0 \\ 0.951 & 0.93 & 0.9 \end{pmatrix} \cdot \begin{pmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \Delta u(t+2|t) \end{pmatrix} \quad (8.14)$$

$$+ \begin{pmatrix} 1.7 & -0.7 \\ 2.19 & -1.19 \\ 2.533 & -1.533 \end{pmatrix} \cdot \begin{pmatrix} y(t) \\ y(t-1) \end{pmatrix} + \begin{pmatrix} -0.6 \\ -1.02 \\ -1.314 \end{pmatrix} \cdot \Delta u(t-1)$$

Por tanto, la matriz del controlador es

$$(G^T G + \lambda_3)^{-1} G^T = \begin{pmatrix} 0.8947 & 0.0929 & 0.0095 \\ -0.8316 & 0.8091 & 0.0929 \\ -0.0766 & -0.8316 & 0.8947 \end{pmatrix} \cdot \Delta u(t-1) \quad (8.15)$$

y la ley de control que se obtiene es

$$\begin{aligned} \Delta u(t) &= 0.644\Delta u(t-1) - 1.7483y(t) + 0.7513y(t-1) \\ &\quad + 0.8947r(t+1) + 0.0929r(t+2) + 0.0095r(t+2) \end{aligned} \quad (8.16)$$

Este es un controlador lineal e invariante en el tiempo, ya que el modelo es invariante. Se puede comprobar la capacidad del controlador de 'mirar hacia adelante', ya que son tenidos en cuenta tres valores futuros de la referencia, tantos como la longitud de la ventana de predicción.

8.4 Control Predictivo RGO.

La mayoría de los procesos industriales se pueden aproximar por procesos de orden uno con un cierto retardo. Esto es debido a que si una constante de tiempo es mucho mayor que las restantes, las más pequeñas trabajan juntas para producir un retardo.

$$G(s) = \frac{k}{1 + \tau s} e^{-s\tau_d} \quad (8.17)$$

Si discretizamos esta función de transferencia obtenemos:

$$G(z^{-1}) = \frac{bz^{-1}}{1 - az^{-1}} z^{-d} \quad (8.18)$$

Considerando un modelo CARIMA (Controlled Auto-Regressive and Integrated Moving-Average) obtenemos

$$(1 - az^{-1})y(t) = bz^{-d}u(t-1) + \frac{\varepsilon(t)}{\Delta} \quad (8.19)$$

donde hemos tomado el polinomio correspondiente al ruido, de grado uno, y donde $u(t)$ e $y(t)$ son las señales de entrada y de salida del sistema, $\varepsilon(t)$ es un ruido blanco y Δ es un integrador ($\Delta = 1 - z^{-1}$). Esta ecuación se puede transformar en

$$y(t+1) = (1+a)y(t) - ay(t-1) + b\Delta u(t-d) + \varepsilon(t+1) \quad (8.20)$$

y obtener la predicción

$$\tilde{y}(t+1) = (1+a)y(t) - a(y(t-1) + b[u(t-d) - u(t-d-1)]) \quad (8.21)$$

A continuación, lo que deseamos hacer es aplicar una secuencia de control que minimice la función objetivo:

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} \delta(j) [\tilde{y}(t+j|t) - r(t+j)]^2 + \sum_{j=1}^{N_2-d} \lambda(j) [\Delta u(t+N-1)]^2 \right\} \quad (8.22)$$

donde $N_1 > d$ y $N_2 = d + N$. Minimizando $J(N)$ respecto a $\Delta u(t)$, $\Delta u(t+1)$, ..., $\Delta u(t+N-1)$, obtenemos

$$M u = P y + R w \quad (8.23)$$

con

$$u = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)]^T \quad (8.24)$$

$$y = [\tilde{y}(t+d|t), \tilde{y}(t+d+1|t)]^T \quad (8.25)$$

$$w = [r(t+d+1), r(t+d+2), \dots, r(t+d+N)]^T \quad (8.26)$$

M y R son matrices de $N \times N$ y P de dimensión $N \times 2$.

Llamando q a la primera fila de la matriz $M-1$, $\Delta u(t)$ viene dada por

$$\Delta u(t) = q P y + q R w \quad (8.27)$$

Por consiguiente el incremento de la señal de control $\Delta u(t)$ resulta:

$$\Delta u(t) = l_{y1} \tilde{y}(t+d|t) + l_{y2} \tilde{y}(t+d-1|t) + l_{r1} r(t) \quad (8.28)$$

donde l_{y1} , l_{y2} y l_{r1} son coeficientes. Por tanto nuestra estrategia es aplicar sucesivamente las ecuaciones:

$$\tilde{y}(t+i) = (1+a)\tilde{y}(t+i-1|t) - a\tilde{y}(t+i-2) + b[u(t+i-d+1) - u(t+i-d)] \quad (8.29)$$

$$u(t) = u(t-1) + (l_{y1} \tilde{y}(t+d|t) + l_{y2} \tilde{y}(t+d-1|t) + l_{r1} r(t))/G \quad (8.30)$$

Es decir:

$$\tilde{y}(t+i) = (1+a)\tilde{y}(t+i-1|t) - a\tilde{y}(t+i-2) + b[u(t+i-d+1) - u(t+i-d)] \quad (8.31)$$

$$u(t) = u(t-1) + (l_{y1} \tilde{y}(t+d|t) + l_{y2} \tilde{y}(t+d-1|t) + l_{r1} r(t))/G \quad (8.32)$$

Esto significa que se puede expresar la ley de control en función de los dos valores estimados de la salida $\tilde{y}(t+d|t)$, $\tilde{y}(t+d-1|t)$ y de la referencia $r(t)$ con tres parámetros desconocidos l_{y1} , l_{y2} , l_{r1} . Si se pueden identificar estos parámetros, tendremos una ley de control. Por supuesto, nos interesa encontrar estos coeficientes de modo que minimicen el funcional 8.22.

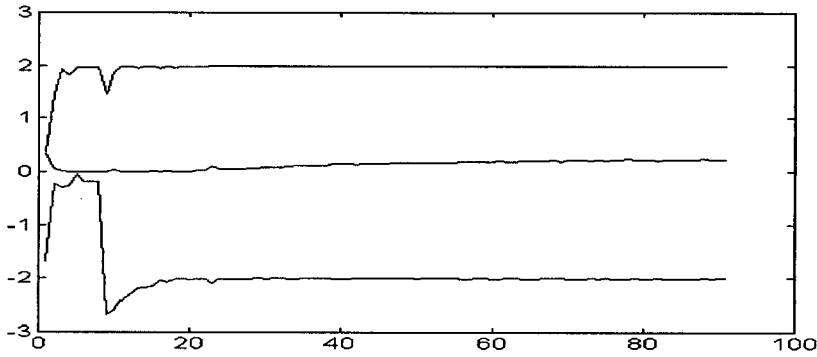


Fig. 8.4: Resultados de la identificación de los parámetros l_{y1} , l_{y2} y l_{r1} del control predictivo RGO.

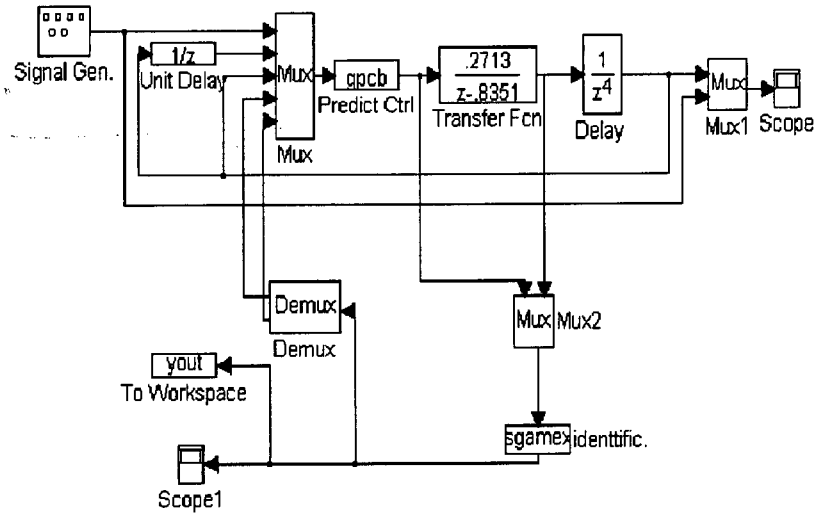


Fig. 8.5: Esquema para la identificación de los parámetros de la ley de control predictivo.

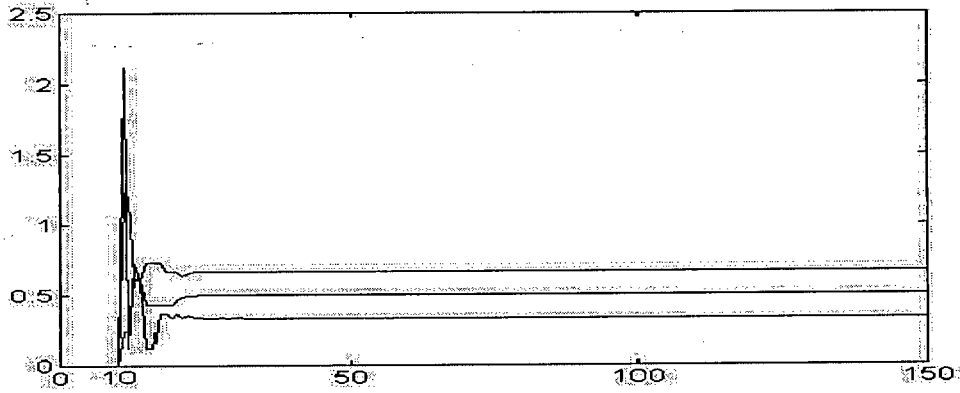


Fig. 8.6: Resultados de la identificación de los parámetros l_{y1} , l_{y2} y l_{r1} del control predictivo RGO. El tiempo 10 corresponde al llenado del buffer.

8.4.1 Ventajas de utilizar RGO en un GPC.

Los algoritmos genéticos optimizan el funcional

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} \delta(j) [\tilde{y}(t+j|t) - r(t+j)]^2 + \sum_{j=1}^{N_2-d} \lambda(j) [\Delta u(t+N-1)]^2 \right\} \quad (8.33)$$

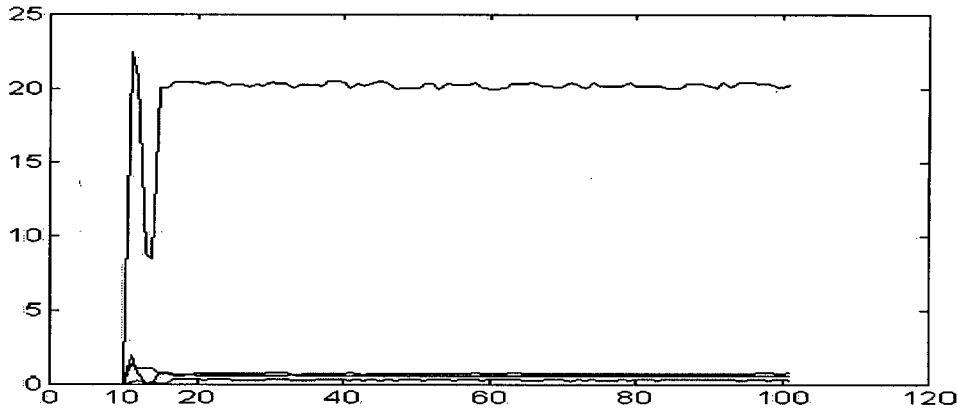


Fig. 8.7: Resultados de la identificación de los parámetros del sistema.

de un modo directo, obteniéndose los coeficientes “ly” y “lr” de la expresión

$$\Delta u(i) = \sum_{k=d}^N l y_k y(i-k) + \sum_{k=1}^{N-d} l r_k r(i-k) \quad (8.34)$$

Lo cual significa que podemos calcular el valor de la entrada del sistema con muy poco coste computacional.

Existen, por tanto, dos bucles: uno lento que nos va actualizando los coeficientes de la señal de entrada u, y un bucle rápido que calcula la señal de entrada. De esta manera se evita el tener que resolver el sistema $Mu=qRy+qRw$ todas las veces, con lo que se gana en velocidad.

8.5 Control Predictivo PID-RGO

8.5.1 Introducción

La sintonización de controles PID digitales para las plantas industriales puede ser difícil algunas veces y consumir bastante tiempo. La situación es mucho peor si el comportamiento de la planta varía con el tiempo.

Las técnicas de algoritmos genéticos off-line para sistemas que no varían con el tiempo fueron propuestas por Porter y Jones [91] como una alternativa posible a la sintonización de los PID digitales. De hecho, se comprobó que los algoritmos genéticos nos permiten una aproximación más sencilla a la sintonización de tales controladores que la solución basada en algoritmos de optimización no genéticos como los propuestos por Polak y Mayne [90] y por Gesing y Davidson [33].

Nuestra técnica permite la sintonización on-line de controladores PID, dotándoles de características del control predictivo muy interesantes como son un menor esfuerzo de control (la señal de control es más suave) y un mejor seguimiento de la señal de referencia.

Recordemos que el esquema general para el control predictivo es:

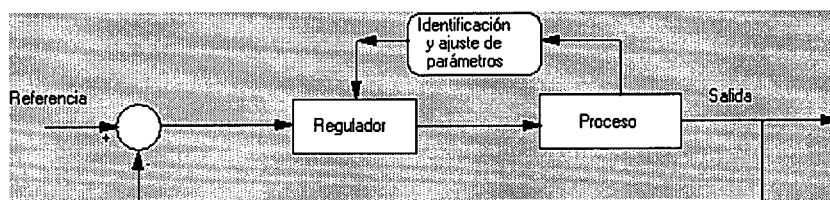


Fig. 8.8: Esquema general del control adaptativo.

El bloque de Identificación y ajuste de parámetros lo dividiremos en dos partes: un primer bloque para identificar el sistema y un segundo bloque para ajustar los parámetros del PID.

Así obtendremos un control adaptativo basado en la optimización de un funcional y en la identificación del proceso.

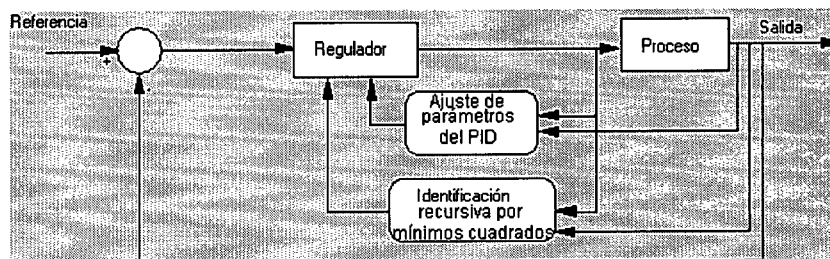


Fig. 8.9:

8.5.2 Identificación mediante Modelos Paramétricos Lineales

Partimos de un modelo de sistema dinámico del tipo:

$$A(q^{-1})y(t) = B(q^{-1})u(t) + e(t) \quad (8.35)$$

donde los polinomios A y B están definidos por:

$$B(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \dots + b_{nb} q^{-nb} \quad (8.36)$$

$$A(q^{-1}) = a_0 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{na} q^{-na} \quad (8.37)$$

Para hacer la identificación de los parámetros es conveniente escribir la ecuación como:

$$y(t) = \phi(t)\theta(t) \quad (8.38)$$

siendo θ el vector de parámetros desconocidos, definido por:

$$\theta^T = [-a_1, -a_2, \dots, -a_{na}, b_0, b_1, b_2, \dots, b_{nb}] \quad (8.39)$$

siendo $\phi(t)$ el vector de regresión que consiste en mediciones entrada/salida dado por:

$$\phi^T(t) = [y(t-1), y(t-2), \dots, y(t-na), u(t-1), u(t-2), \dots, u(t-nb)] \quad (8.40)$$

La identificación de un sistema dinámico se puede hacer off-line, o bien on-line. En el primer caso uno de los métodos mas clásicos es el método de mínimos cuadrados, cuya solución viene dada por

$$\theta(t) = [\Phi^T \Phi]^{-1} [\Phi^T Y] \quad (8.41)$$

siendo la matriz Φ es la matriz de observaciones, cuyas filas están formadas por el vector definido anteriormente e Y es el vector de salidas, ambas con N filas, correspondientes a las N últimas ecuaciones $y(t) = \phi(t)\theta, \dots, y(t-n+1) = \phi(t-n+1)\theta$ escritas en forma vectorial. Sin embargo es mas útil en términos de control, realizar la identificación on-line de modo recursivo. Si en la ecuación anterior denominamos $P(t) = [\Phi^T \Phi]^{-1}$ entonces

$$P(t)^{-1} = \left[\sum_{s=1}^t \phi(s)\phi^T(s) \right] = \left[\sum_{s=1}^{t-1} \phi(s)\phi^T(s) + \phi(t)\phi^T(t) \right] = P(t-1)^{-1} + \phi(t)\phi^T(t) \quad (8.42)$$

Análogamente, para $\Phi^T Y$ podemos escribir:

$$\phi^T(t)Y(t) = \phi^T(t-1)Y(t-1) + \phi^T(t)Y(t) \quad (8.43)$$

con lo que podemos expresar la ecuación $\theta(t) = [\Phi^T \Phi]^{-1} [\Phi^T Y]$ de forma recursiva como:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t)(y(t) - \phi^T(t)\hat{\theta}(t-1)) \quad (8.44)$$

donde llamamos $e(t) = (y(t) - \phi^T(t)\hat{\theta}(t-1))$. Para obtener $P(t)$ a partir de $P(t-1)$ usamos el teorema de inversión de matrices y se obtiene

$$P(t) = P(t-1) \left[\phi(t) - \frac{\phi(t)\phi^T(t)P(t-1)}{1 + \phi^T(t)P(t-1)\phi(t)} \right] \quad (8.45)$$

Con todo ello podemos estimar los parámetros de forma recursiva como:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)e(t) \quad (8.46)$$

8.5.3 Control Predictivo PID-RGO

El control predictivo basado en modelo utiliza el modelo del sistema obtenido de la identificación para incorporar el comportamiento futuro en el procedimiento de diseño del controlador. Se parte del modelo CARIMA del sistema,

$$A(q^{-1})y(t)B(q^{-1})u(t-1) + \frac{C(q^{-1})}{1-q^{-1}}\varepsilon(t) \quad (8.47)$$

y se halla la secuencia de control que minimiza una función de coste de la forma:

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} (\hat{y}(t+j|t) - r(t+j))^2 + \lambda \sum_{j=1}^{N_u} (\Delta u(t+j-1)) \right\} \quad (8.48)$$

donde E es la esperanza matemática, $\hat{y}(t+j|t)$ es una secuencia de j predicciones de la salida del sistema posteriores al instante t ; N_1 y N_2 son los horizontes de predicción, N_u es el horizonte de control y $\lambda(j)$ es una secuencia de ponderación.

En nuestro caso optimizamos este funcional por medio de algoritmos genéticos y así conseguimos una secuencia de control futura $u(t)$, $u(t+1)$,...de modo que la salida $y(t+j)$ se aproxime a la referencia $r(t+j)$. Deseamos hallar los coeficientes del controlador PID que optimizan el funcional anterior, para lo cual necesitamos una expresión que nos dé los valores de $\delta u(t+j-1)$ en función de los valores de $\hat{y}(t+j|t) - r(t+j)$ y de los parámetros del PID. La ecuación del PID será

$$u(t) = \left[K_p + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1}) \right] (r(t-1) - y(t-1)) \quad (8.49)$$

por tanto

$$\Delta u(t) = [K_P(1 - z^{-1}) + K_I + K_D(1 - z^{-1})^2](r(t-1) - y(t-1)) \quad (8.50)$$

$$\begin{aligned} \Delta u(t) &= (K_P + K_I + K_D)(r(t-1) - y(t-1)) \\ &\quad + (K_P - 2K_D)(r(t-2) - y(t-2)) + (-K_D)(r(t-3) - y(t-3)) \end{aligned} \quad (8.51)$$

$$\begin{aligned} \Delta u(t) &= C_1(y(t-1) - r(t-1)) + C_2(y(t-2) - r(t-2)) \\ &\quad + C_3(y(t-3) - r(t-3)) \end{aligned} \quad (8.52)$$

De esta manera, el funcional queda en función de los valores $\hat{y}(t+j|t) - r(t+j)$ y de los coeficientes C_1, C_2, C_3 . Por tanto, podemos hallar los coeficientes C_1, C_2, C_3 que optimizan el funcional y después hallar los coeficientes del PID correspondiente:

$$\begin{cases} K_P = -C_2 + 2C_3 \\ K_I = -C_1 + C_2 - 3C_3 \\ K_D = C_3 \end{cases} \quad (8.53)$$

8.5.4 Simulación del control adaptativo PID-RGO.

Deseamos comprobar el funcionamiento del controlador RGO-predictivo para varios valores de λ en el funcional 8.48, el peso del esfuerzo de control. Para valores pequeños de λ tendremos que el esfuerzo de control es grande y por tanto el control será más rápido.

A partir de las bibliotecas de bloques realizamos el esquema siguiente:

En las figuras 8.11, 8.12 y 8.13 se puede comprobar el comportamiento del sistema para varios valores de λ : 0.12, 0.16 y 0.2.

El factor de peso de control λ afecta a la señal de control. Cuanto mayor sea su valor, menor esfuerzo de control. Si se le da un valor pequeño la respuesta del sistema será más rápida puesto que el sistema tiende a minimizar el error entre la salida y la referencia, olvidando el esfuerzo de control. Los valores de los parámetros de control l_{y1} , l_{y2} y l_{r1} y por tanto los polos en bucle cerrado dependen de los valores de λ .

8.6 Análisis de resultados

A la vista de los análisis y simulaciones realizados en las secciones anteriores, se desprenden los siguientes resultados:

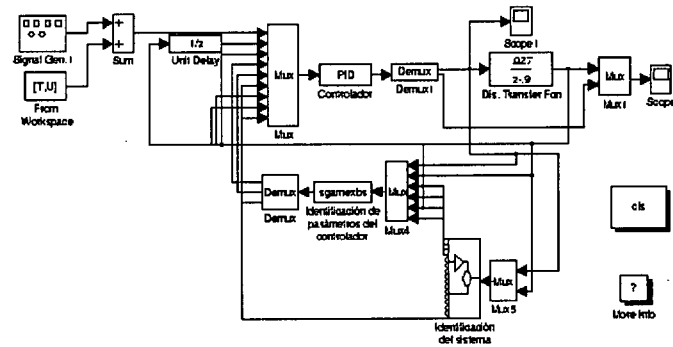


Fig. 8.10: Diagrama Simulink para Control 1

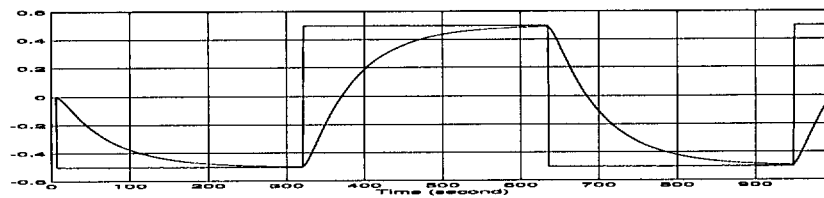


Fig. 8.11: Resultados del Control Predictivo con RGO para $\lambda = 0.12$

- La optimización genética restringida (RGO) proporciona un marco unificador, muy flexible y natural, sobre el que desarrollar nuevos controladores predictivos. La naturaleza de esto es controladores puede ser muy variada, en particular, haciendo ciertas aproximaciones, se tiene controladores de tipo PID adaptativos.
- A diferencia de otras técnicas de optimización, la optimización genética restringida hace una gestión sencilla e intuitiva de las ligaduras.

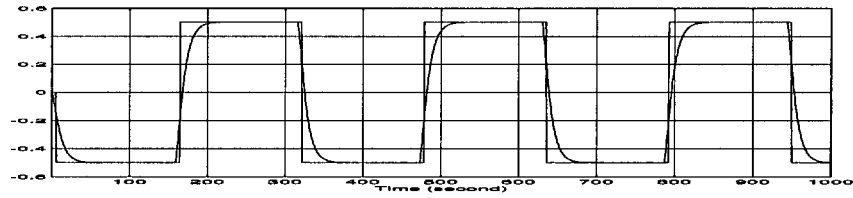


Fig. 8.12: Resultados del Control Predictivo con RGO para $\lambda = 0.16$

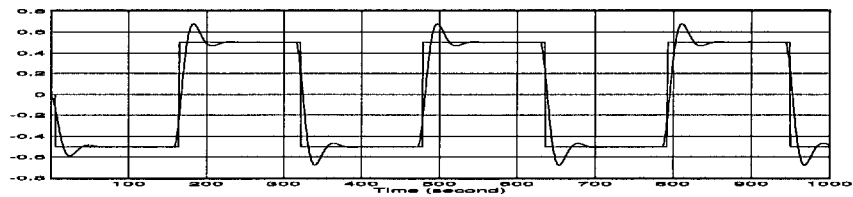


Fig. 8.13: Resultados del Control Predictivo con RGO para $\lambda = 0.20$

- Este método se puede aplicar a cualquier sistema causal, incluyendo a los sistemas no lineales variantes en el tiempo, siempre que las no linealidades no sean demasiado grandes.
- Como problema, se puede apuntar la complejidad computacional del método, lo que impide variar la ley de control muy rápidamente (del orden de 0.1 s).

9. APLICACIONES I: TRMS

9.1 Introducción

En este capítulo se considera la aplicación a un sistema real: el Sistema MIMO de Rotores Gemelos (Twin Rotor MIMO System - TRMS), que es un laboratorio diseñado expresamente para realizar experimentos de control. En ciertos aspectos su comportamiento recuerda al de un helicóptero. Desde el punto de vista de control ejemplifica un sistema de orden alto con acoplamientos cruzados muy significativos.



Fig. 9.1: *Twin Rotor MIMO System (TRMS).*

El TRMS consiste en un eje pivotado en su base de forma que puede rotar libremente en el plano horizontal y vertical. En las terminaciones de la barra hay dos rotores (el principal y el de cola) dotados con motores de corriente continua. Una barra con contrapeso está fijada al eje en el punto en que pivota.

El estado del eje se puede describir por cuatro variables de estado: los ángulos horizontal y vertical medidos por sensores de posición y dos correspondientes a las velocidades angulares. Dos variables de estado adicionales son las velocidades

angulares de los rotores, medidas por tacogeneradores acoplados a los motores de continua.

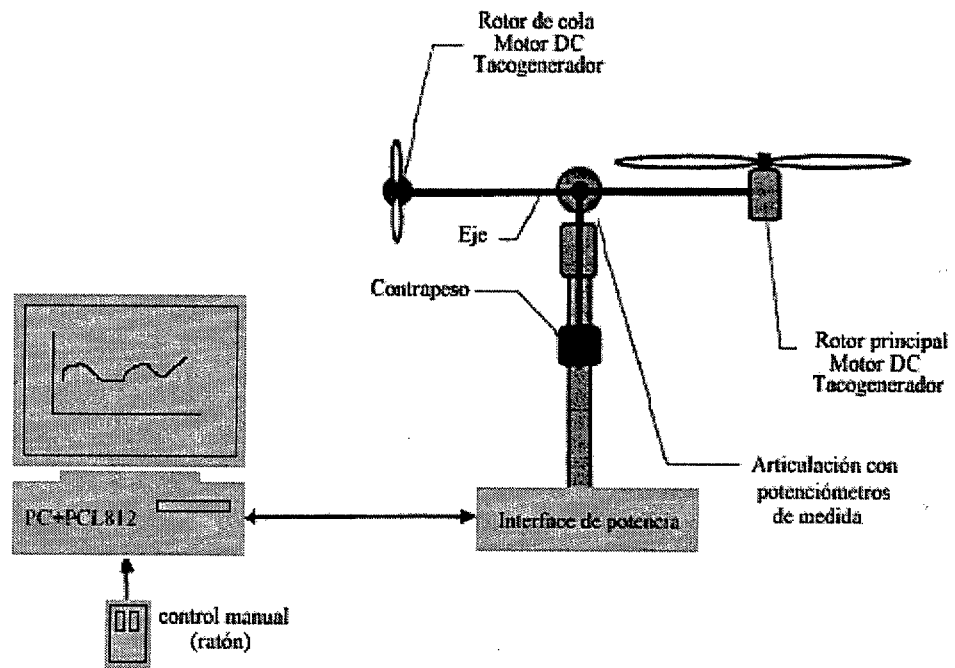


Fig. 9.2: *Twin Rotor MIMO System (TRMS).*

En un helicóptero normal la fuerza aerodinámica es controlada cambiando el ángulo de ataque. El TRMS está construido de forma que el ángulo de ataque es fijo. La fuerza aerodinámica se controla variando la velocidad de los rotores. Por tanto, las entradas de control son las tensiones suministradas a los motores de continua. Un cambio en el valor de la tensión hace que cambie la velocidad de rotación lo que hace cambiar la correspondiente posición de la barra.

Sin embargo, se observan acoplamientos cruzados significativos entre las acciones de los rotores; cada rotor influencia las dos posiciones angulares.

El diseño de controladores de éste sistema deben estar basados en desacoplos. Para el sistema desacoplado, se pueden aplicar entradas de control independientes para cada coordenada del sistema.

Se puede utilizar un ordenador PC para el control en tiempo real del TRMS. Para ello se necesita una tarjeta controladora (PCL-812 en nuestro caso).

9.2 Modelado y parámetros

Los métodos modernos de diseño y adaptación de controladores en tiempo real requieren modelos matemáticos de alta calidad del sistema. Para sistemas de orden alto, no lineales y con acoplamientos los métodos clásicos de modelado (basados en ecuaciones de Lagrange) son, a menudo, muy complicados. Por eso hemos decidido utilizar una aproximación más sencilla, basada en la representación en diagramas de bloques del sistema, y que es muy útil para su utilización en el entorno SIMULINK.

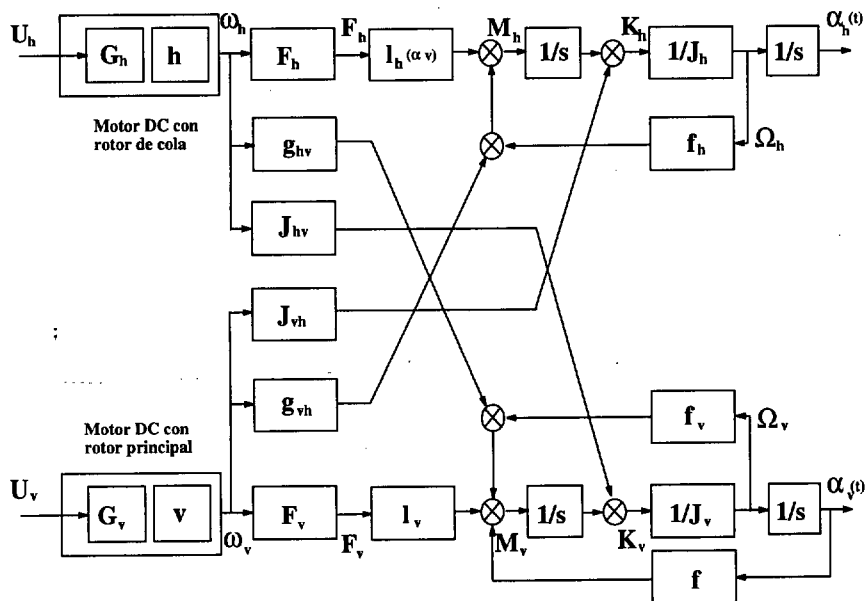


Fig. 9.3: Diagrama de bloques del modelo TRMS.

El diagrama de bloques se muestra en la figura. Las tensiones de control U_h y U_v son las entradas de los motores de continua que mueven los rotores. El modelo de motor de continua con propulsor se compone de un sistema con dinámica lineal seguido de una no linealidad estática. La parte lineal tiene forma de función de transferencia de primer orden $G_h = \frac{1}{T_h s + 1}$ y $G_v = \frac{1}{T_v s + 1}$. Las funciones no lineales $h(U_h)$ y $v(U_v)$ son las características estáticas de los motores de continua con propulsores. La tensión de entrada está limitada al rango de $\pm 10v$. Las relaciones no lineales entre la velocidad de los rotores y la fuerza aerodinámica resultante puede ser aproximada por las relaciones cuadráticas: $F_h = \text{sign}(\omega_h) k_h \omega_h^2$ y $F_v = \text{sign}(\omega_v) k_v \omega_v^2$ (k_h y k_v constantes positivas).

Es importante tener en cuenta que la forma geométrica de los propulsores no es simétrica, y por lo tanto el comportamiento en una dirección de diferente del correspondiente a la otra dirección.

Tab. 9.1: Notación usada en la figura 9.2

Símbolo	Significado	Unidades
α_h	posición horizontal del eje del TRMS (azimut position)	[rad]
Ω_h	velocidad angular del eje del TRMS (azimut velocity)	[rad/s]
U_h	entrada de control (tensión) del motor horizontal	[V]
G_h	función de transferencia lineal del motor y rotor de cola	
h	parte no-lineal del motor y rotor de cola: $h(U_h) = \omega_h$	[rad/s]
ω_h	velocidad de rotación del rotor de cola	[rad/s]
F_h	función no-lineal (cuadrática) de fuerza aerodinámica del rotor de cola: $F_h = F_h(\omega_h)$	[N]
l_h	brazo efectivo de la fuerza aerodinámica del rotor de cola: $l_h = l_h(\alpha_v)$	[m]
J_h	función no-lineal del momento de inercia respecto al eje vertical: $J_h = J_h(\alpha_v)$	[kgm ²]
M_h	par de giro horizontal	[Nm]
K_h	momento angular horizontal	[Nms]
f_h	momento de la fuerza de fricción del eje vertical	[Nm]
α_v	posición vertical del eje del TRMS (pitch position)	[rad]
Ω_v	velocidad angular del eje del TRMS (pitch velocity)	[rad/s]
U_v	entrada de control (tensión) del motor vertical	[V]
G_v	función de transferencia lineal del motor y rotor principal	
V	parte no-lineal del motor y rotor principal: $h(U_v) = \omega_v$	[rad/s]
ω_v	velocidad de rotación del rotor principal	[rad/s]
F_v	función no-lineal (cuadrática) de fuerza aerodinámica del rotor principal: $F_v = F_v(\omega_v)$	[N]
l_v	brazo efectivo de la fuerza aerodinámica del rotor principal	[m]
J_v	momento de inercia respecto al eje horizontal	[kgm ²]
M_v	par de giro vertical	[Nm]
K_v	momento angular vertical	[Nms]
f_v	momento de la fuerza de fricción del eje horizontal	[Nm]
f	par de fuerzas vertical del contrapeso: $f = f(\alpha_v)$	[Nm]

continúa en la página siguiente

viene de la página anterior		
Símb.	Significado	Unidades
J_{hv}	momento angular vertical del rotor de cola	[Nms]
J_{vh}	momento angular horizontal del rotor principal	[Nms]
g_{vh}	función no-lineal (cuadrática) del par de giro de reacción: $g_{vh} = g_{vh}(\omega_v)$	[Nm]
g_{hv}	función no-lineal (cuadrática) del par de giro de reacción: $g_{hv} = g_{hv}(\omega_h)$	[Nm]
t	tiempo	[s]
$1/s$	función de transferencia del integrador	

La rotación del propulsor produce un momento angular que, de acuerdo con la ley de conservación del momento angular, puede ser compensada por las partes restantes del TRMS. Esto hace que exista una interacción entre las dos funciones de transferencia, representadas por el momento de inercia de los motores con propulsores J_{hv} y J_{vh} . Esta interacción influencia directamente a las velocidades de los ejes en ambos planos. Los pares de fuerza que actúan son las fuerzas F_h y F_v multiplicadas por la longitud del brazo $l_h(\alpha_v)$ y l_v .

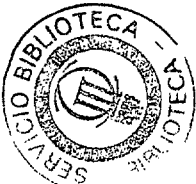
Controlar el sistema consiste en estabilizar el eje del TRMS en una posición arbitraria deseada (pitch y azimut), dentro de unos límites prácticos, o bien, hacer que siga una trayectoria deseada.

9.3 Identificación del TRMS

Antes de estudiar los diversos controladores para el TRMS, vamos a realizar un primer estudio del sistema para saber cuáles son los problemas que nos podemos encontrar. Para ello estudiamos la respuesta ante una onda cuadrada del rotor principal. Se puede apreciar en la figura que por debajo de la posición de reposo (la de equilibrio cuando el motor está parado) se comporta prácticamente como un oscilador puro, mientras que por encima de la posición de reposo se comporta como un oscilador amortiguado.

El rotor secundario por su parte no tiene una fuerza que se contraponga a la fuerza de control, como ocurre con la gravedad en el otro rotor y además apenas tiene rozamiento. Esto motiva que sea necesario cambiar el sentido de giro de las aspas continuamente para estar en un punto fijo.

Antes de intentar hacer una identificación adaptativa, vamos a ver cómo funciona una identificación off-line.



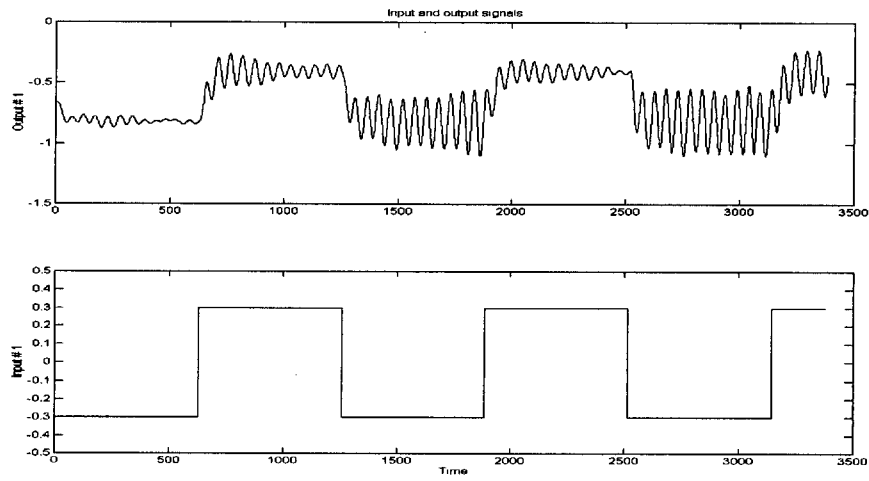


Fig. 9.4: Respuesta del rotor principal del TRMS a una onda cuadrada.

9.4 Control PID del TRMS

Para poder realizar un estudio comparativo de los diversos controladores, vamos a ver los resultados que se obtienen al controlar el TRMS con un PID adaptativo RGO.

9.4.1 Estabilización y seguimiento 1-DOF vertical

La tarea de control de estabilización en un grado de libertad (1 DOF) consiste en mover el TRMS a una posición arbitraria en el plano seleccionado y estabilizarlo en esa posición. El seguimiento consiste en que los movimientos del TRMS sigan la señal de referencia lo mas fielmente posible.

Para comenzar restringimos nuestro objetivo de control a la estabilización y seguimiento del sistema en el plano vertical solamente. El correspondiente diagrama de bloques del control PID del sistema se muestra en la figura 9.6.

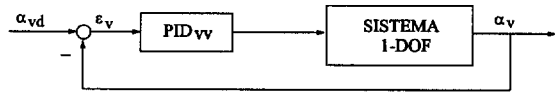


Fig. 9.5: Control del sistema 1-DOF (estabilización y seguimiento en vertical). Arriba: salida vs. señal de referencia. Abajo: error cometido.

El diagrama de bloques siguientes muestra el sistema de una forma más detallada. Nótese que sólo se considera la parte vertical del sistema de control.

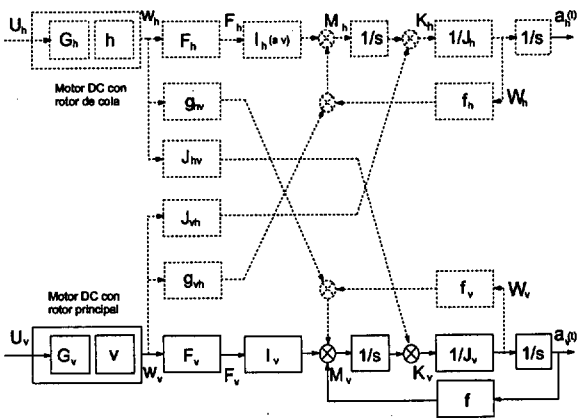


Fig. 9.6: Diagrama de bloques del sistema 1-DOF (plano vertical).

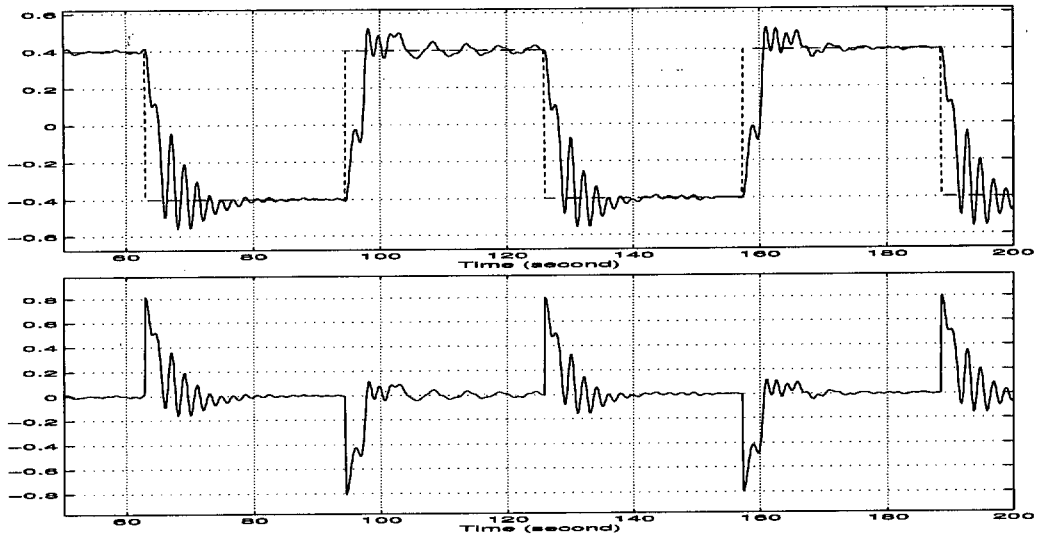


Fig. 9.7: Control del sistema 1-DOF (estabilización y seguimiento en vertical).
Arriba: salida del sistema vs. señal de referencia. Abajo: error cometido.

9.4.2 Estabilización y seguimiento 1-DOF horizontal

La tarea de control de estabilización en un grado de libertad (1 DOF) consiste en mover el TRMS a una posición arbitraria en el plano seleccionado y estabilizarlo en esa posición. Para ello bloqueamos un eje del sistema de forma que no se pueda mover en el plano vertical. El seguimiento consiste en que los movimientos del TRMS sigan la señal de referencia lo mas fielmente posible.

Para comenzar restringimos nuestro objetivo de control a la estabilización y seguimiento del sistema en el plano horizontal solamente. El correspondiente diagrama de bloques del control PID del sistema se muestra en la figura 9.8.

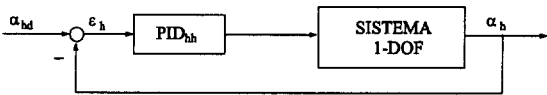


Fig. 9.8: Control del sistema 1-DOF (estabilización y seguimiento horizontal).

El diagrama de bloques siguientes muestra el sistema de una forma más detallada. Nótese que sólo se considera la parte horizontal del sistema de control.

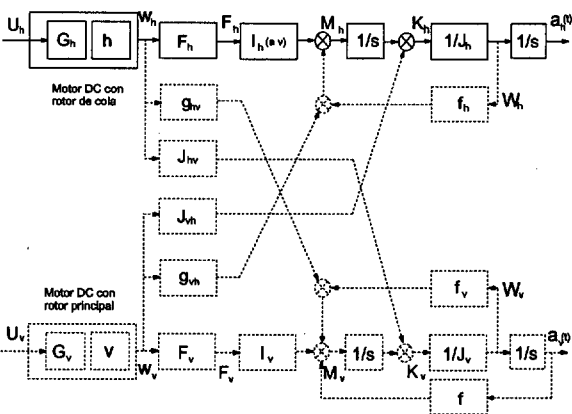


Fig. 9.9: Diagrama de bloques del sistema 1-DOF (plano horizontal).

9.4.3 Estabilización y seguimiento 2-DOF.

La tarea en este caso es la misma que en las secciones previas pero el TRMS no está bloqueado mecánicamente, y por tanto está libre para moverse en ambos planos.

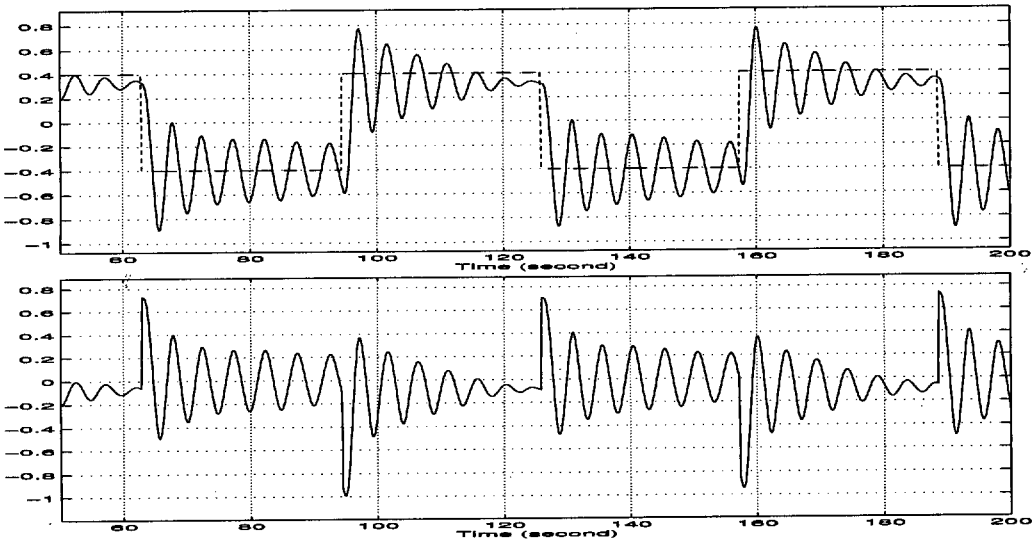


Fig. 9.10: Control del sistema 1-DOF (estabilización y seguimiento en horizontal). Arriba: salida del sistema vs. señal de referencia. Abajo: error cometido.

Controlador PID simple El controlador PID simple controla los movimientos horizontal y vertical separadamente. En este sistema de control la influencia de un rotor sobre el otro no está compensada por la estructura del controlador. El sistema no está desacoplado.

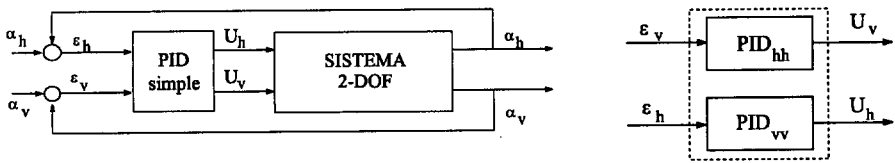


Fig. 9.11: Control del sistema 2-DOF.

9.4.4 Controlador PID con acoplamiento cruzado

El controlador PID con acoplamiento cruzado controla el sistema en los planos horizontal y vertical. En este sistema de control la influencia de un rotor

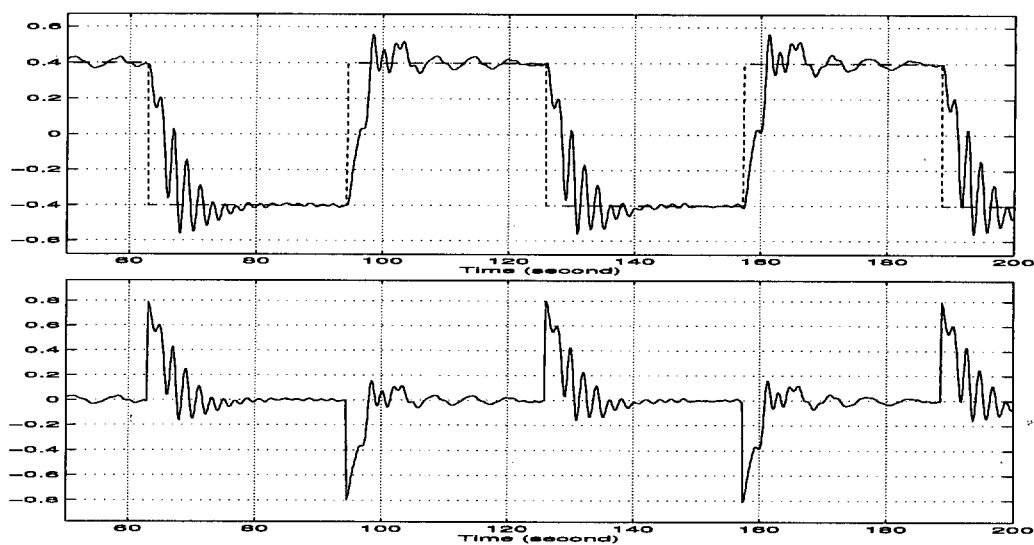


Fig. 9.12: Experimento con el controlador de estabilización y seguimiento 2-DOF PID. Arriba: salida del sistema vs. señal de referencia en eje vertical.
Abajo : error cometido

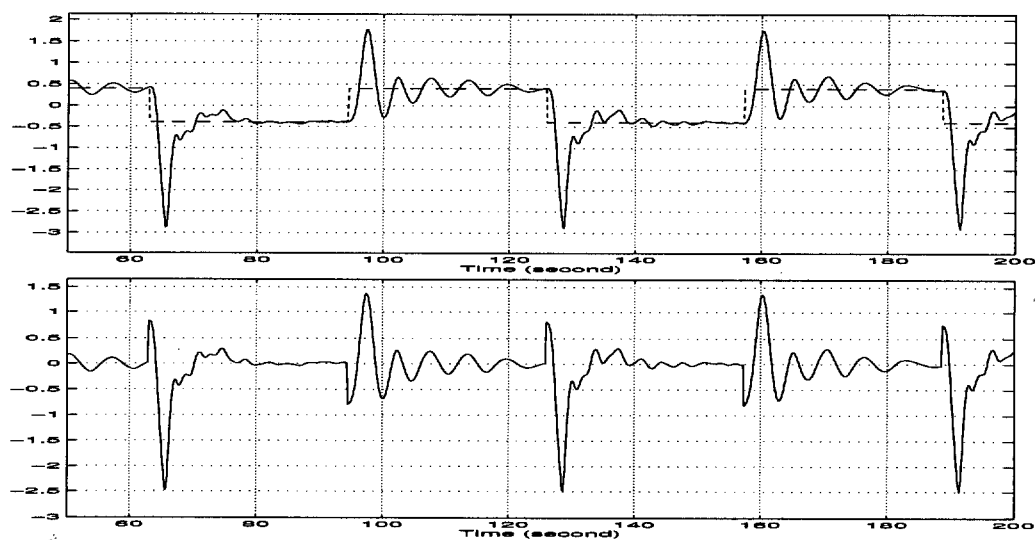


Fig. 9.13: Experimento con el controlador de estabilización y seguimiento 2-DOF PID. Arriba: salida del sistema vs. señal de referencia en eje horizontal.
Abajo : error cometido

sobre el movimiento en el otro plano puede ser compensada por la estructura de acoplamiento cruzado del controlador.

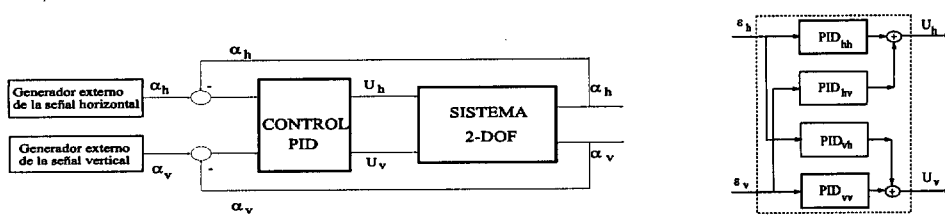


Fig. 9.14: Diagrama de bloques del sistema de control 2-DOF PID de acoplamiento cruzado.

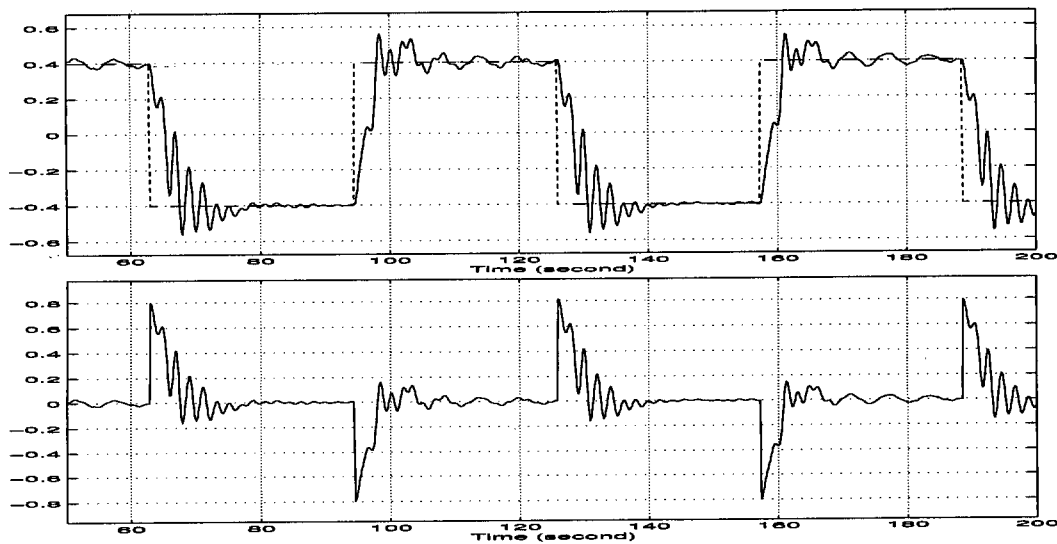


Fig. 9.15: Experimento con el controlador de estabilización y seguimiento 2-DOF PID con acoplamiento cruzado. Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo : error cometido

9.5 Control Predictivo mediante PID y RGO

El control predictivo basado en modelo utiliza el modelo del sistema obtenido de la identificación para incorporar el comportamiento futuro en el procedimiento de diseño del controlador. Se parte del modelo CARIMA del sistema,

$$A(q^{-1})y(t)B(q^{-1})u(t-1) + \frac{C(q^{-1})}{1-q^{-1}}\varepsilon(t) \quad (9.1)$$

y se halla la secuencia de control que minimiza una función de coste de la forma:

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} (\hat{y}(t+j|t) - r(t+j))^2 + \sum_{j=1}^{N_u} (\Delta u(t+j-1)) \right\} \quad (9.2)$$

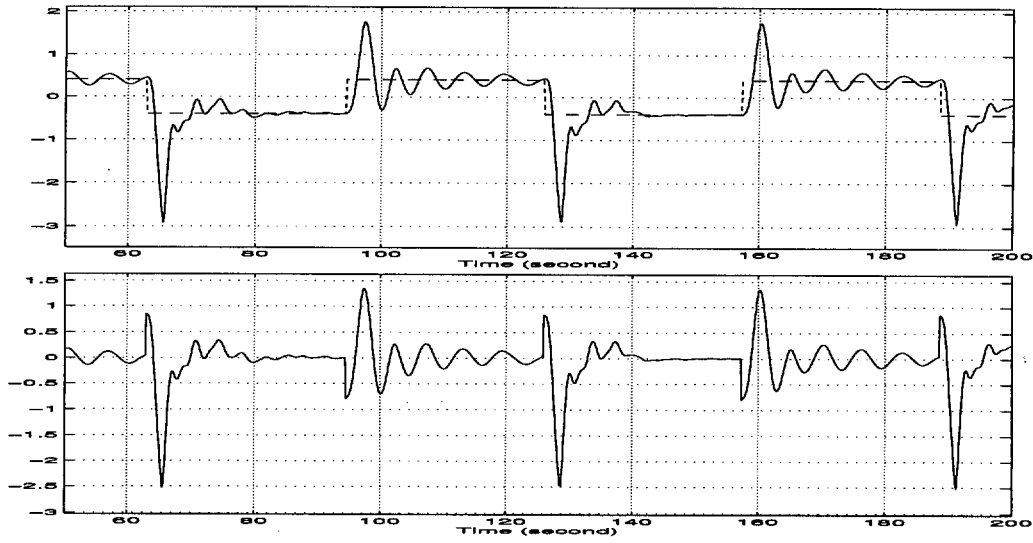


Fig. 9.16: Experimento con el controlador de estabilización y seguimiento 2-DOF PID con acoplamiento cruzado. Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo : error cometido.

donde E es la esperanza matemática, $\hat{y}(t+j|t)$ es una secuencia de j predicciones de la salida del sistema posteriores al instante t ; N_1 y N_2 son los horizontes de predicción, N_u es el horizonte de control y $\lambda(j)$ es una secuencia de ponderación.

En nuestro caso optimizamos este funcional por medio de algoritmos genéticos y así conseguimos una secuencia de control futura $u(t)$, $u(t+1)$,... de modo que la salida $y(t+j)$ se aproxime a la referencia $r(t+j)$. Deseamos hallar los coeficientes del controlador PID que optimizan el funcional anterior, para lo cual necesitamos una expresión que nos dé los valores de $\delta u(t+j-1)$ en función de los valores de $\hat{y}(t+j|t) - r(t+j)$ y de los parámetros del PID. La ecuación del PID será

$$u(t) = \left[K_p + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1}) \right] (r(t-1) - y(t-1)) \quad (9.3)$$

por tanto

$$\Delta u(t) = [K_p(1-z^{-1}) + K_I + K_D(1-z^{-1})^2](r(t-1) - y(t-1)) \quad (9.4)$$

$$\begin{aligned} \Delta u(t) = & (K_p + K_I + K_D)(r(t-1) - y(t-1)) \\ & + (K_p - 2K_D)(r(t-2) - y(t-2)) + (-K_D)(r(t-3) - y(t-3)) \end{aligned} \quad (9.5)$$

$$\Delta u(t) = C_1(y(t-1) - r(t-1)) + C_2(y(t-2) - r(t-2)) + C_3(y(t-3) - r(t-3)) \quad (9.6)$$

De esta manera, el funcional queda en función de los valores $\hat{y}(t+j|t) - r(t+j)$ y de los coeficientes C_1, C_2, C_3 . Por tanto, podemos hallar los coeficientes C_1, C_2, C_3 que optimizan el funcional y después hallar los coeficientes del PID correspondiente:

$$\begin{cases} K_P = -C_2 + 2C_3 \\ K_I = -C_1 + C_2 - 3C_3 \\ K_D = C_3 \end{cases} \quad (9.7)$$

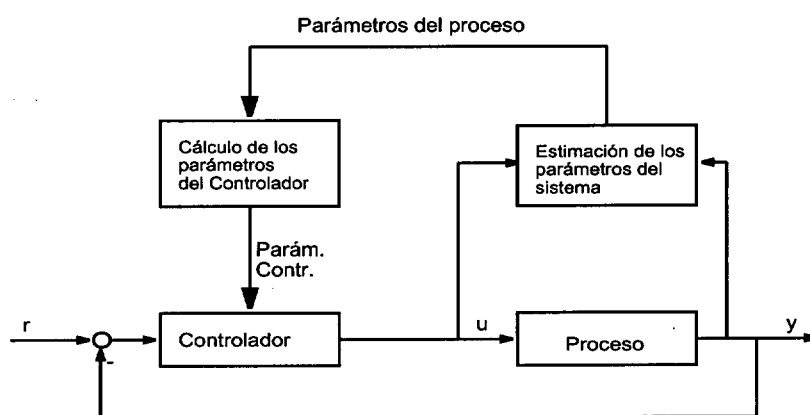


Fig. 9.17: Sistema adaptativo con identificación de modelo (MIAS)

9.6 Control predictivo del TRMS

Basándose en los estudios realizados en los capítulos anteriores se ha desarrollado un controlador predictivo que utiliza las técnicas RGO en dos niveles: para identificar un modelo del sistema y para encontrar los parámetros del controlador.

Obviamente, un control adaptativo y predictivo necesita una gran cantidad de trabajo computacional. En cada instante de muestreo los parámetros del proceso deben ser identificados, el modelo predictivo debe ser calculado y la función de coste debe ser evaluada. Estas operaciones consumen tiempo y pueden hacer que la señal de control no se consiga variar con la rapidez necesaria.

Para la realización del controlador se optimiza el funcional

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - r(t+j)]^2 + \sum_{j=1}^{N_2-d} \lambda(j) [\Delta u(t+N-1)]^2 \right\} \quad (9.8)$$

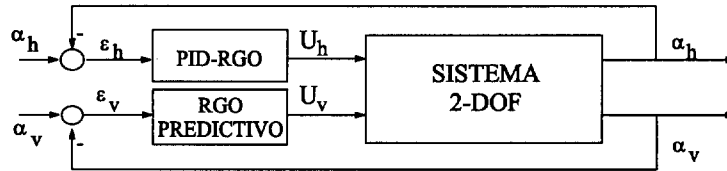


Fig. 9.18: Control del sistema con RGO-PID y RGO Predictivo.

de un modo directo, utilizando la Optimización Genética Restringida, obteniéndose los coeficientes “ly” y “lr” de la expresión

$$\Delta u(i) = \sum_{k=d}^N l y_k y(i-k) + \sum_{k=1}^{N-d} l r_k r(i-k). \quad (9.9)$$

Lo cual significa que podemos calcular el valor de la entrada del sistema con poco coste computacional.

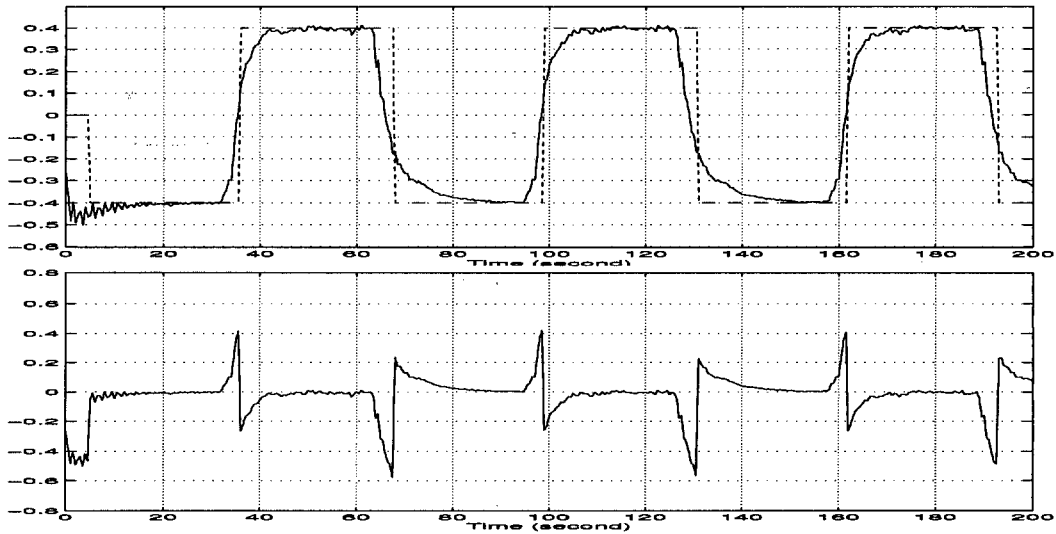


Fig. 9.19: Experimento con el controlador de estabilización y seguimiento: control predictivo-RGO en el plano vertical (con un grado de libertad). Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo: error cometido.

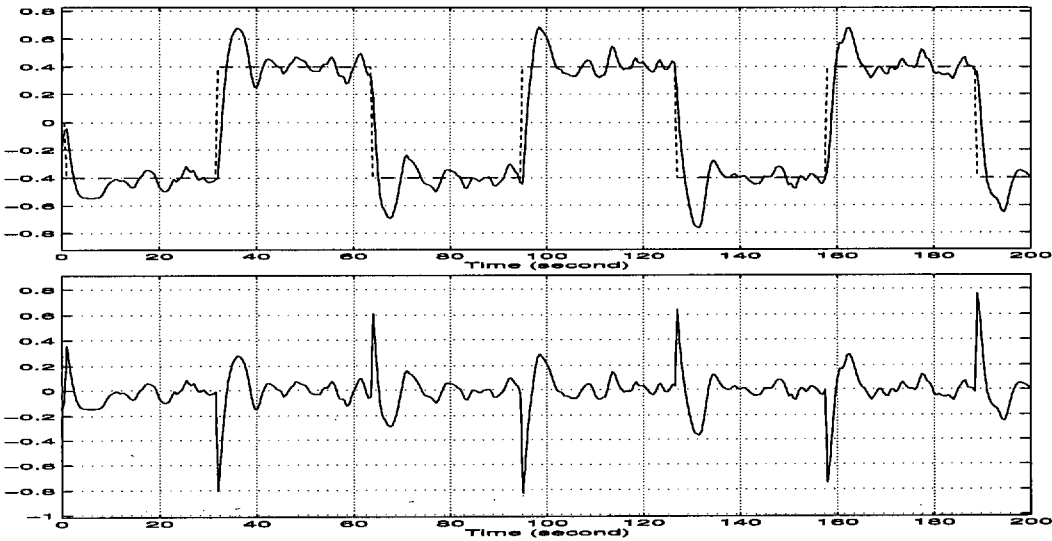


Fig. 9.20: Experimento con el controlador de estabilización y seguimiento: PID-RGO (arriba) en el plano vertical (con un grado de libertad). Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo : error cometido.

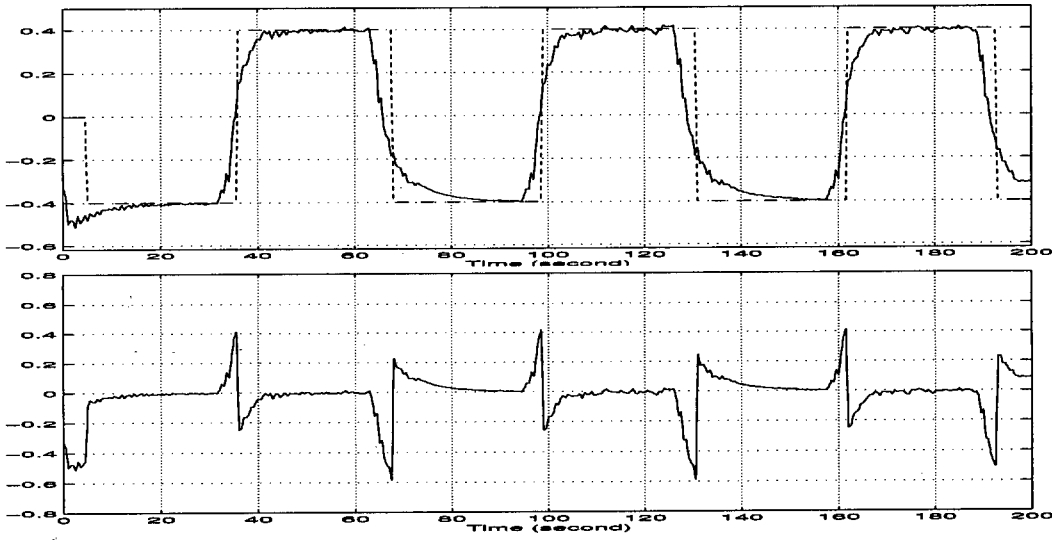


Fig. 9.21: Experimento con el controlador de estabilización: Predictivo-RGO con dos grados de libertad. Arriba: salida del sistema vs. señal de referencia en eje vertical. Abajo: error cometido.



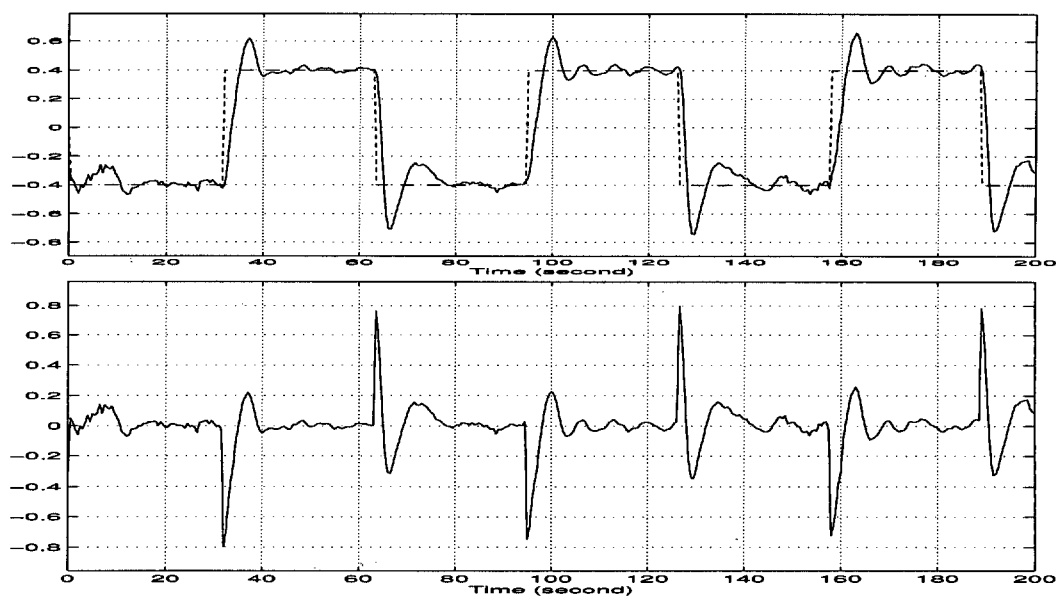


Fig. 9.22: Experimento con el controlador de estabilización: PID-RGO con dos grados de libertad. Arriba: salida del sistema vs. señal de referencia en eje horizontal. Abajo: error cometido.

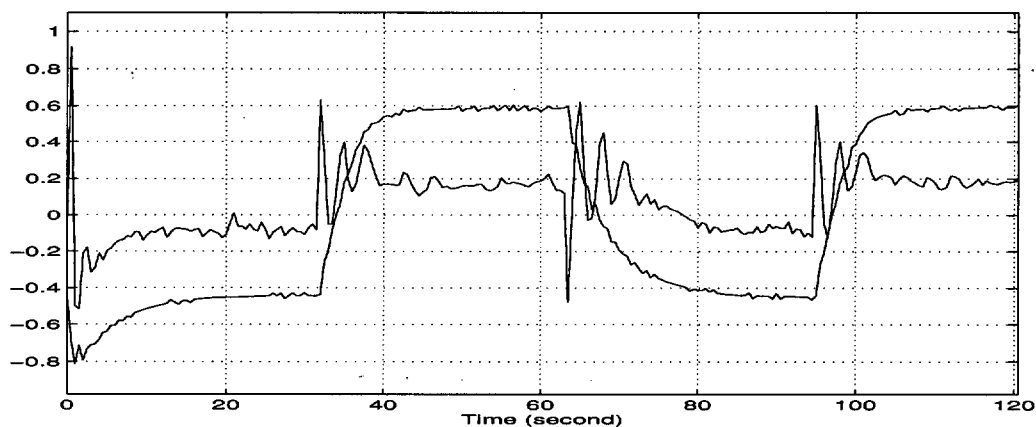


Fig. 9.23: Experimento con el controlador de estabilización (señales de control: control predictivo y PID).

Existen, por tanto, dos bucles: uno lento que nos va actualizando los coeficientes de la señal de entrada u , y un bucle rápido que calcula la señal de entrada. De esta manera se evita el tener que resolver el sistema $Mu = qRy + qRw$ todas las veces, con lo que se gana en velocidad.

Esto nos permite hacer uso de modelos y de ligaduras altamente no lineales, y de funciones objetivo no cuadráticas, por lo que resultan más potentes y flexibles que los convencionales.

Para poder hacer frente a unas condiciones cambiantes, la etapa de optimización ha de ser resuelta en tiempo real.

9.7 Control en el Espacio de Estados del TRMS

Hemos desarrollado un controlador PID que utiliza las técnicas RGO en dos niveles: para identificar un modelo del sistema y para encontrar los parámetros del controlador.

Como primer paso, realizamos una identificación paramétrica, para lo cual partimos de una primera identificación de múltiples modelos de tipo ARX, $A(q^{-1})y_k = z^{-d}B(q^{-1})u_k + e(t)$, mediante el método de mínimos cuadrados, y después se transforma en una realización en el espacio de estados.

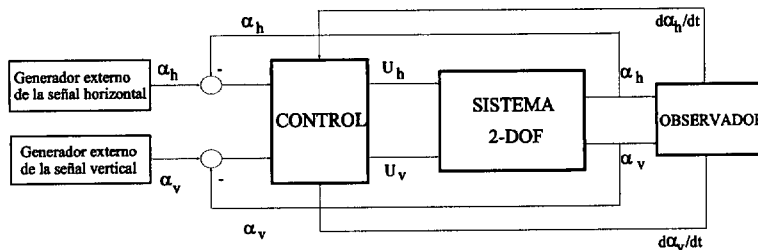


Fig. 9.24: Control del sistema 2-DOF.

A continuación realizamos la búsqueda de modelos que se vayan adaptando mejor al sistema por medio de un algoritmo genético.

Para hacer la función de pérdida $V_n(i) = \sum_{k=1}^{buff} (y_{n-k} - \hat{y}_{n-k})^2$ lo más pequeña posible, se ha utilizado como función de salud $F_n(i) = \frac{1}{cte + V_n(i)}$, donde "cte" es una constante suficientemente grande (10^8) para mantener la diversidad de la población. La probabilidad de reproducción se ha calculado como $F_n(t) = F(t)/\bar{F}$.

En cada generación, sustituimos el anterior modelo por el mejor de la nueva generación como centro del entorno donde se realiza la búsqueda, de esta manera, en caso de un sistema LTI, el mejor modelo debe ir tendiendo al valor que minimiza el funcional y la población iría tendiendo a cero. A continuación realizamos

una nueva lectura de la entrada y la salida del sistema.

De esta manera conseguimos un conjunto de aproximaciones sucesivas a mejores modelos en caso de el sistema sea invariante en el tiempo y si el sistema es variable, pero de variación lo suficientemente lenta, entonces el modelo va siguiendo al sistema en sus variaciones.

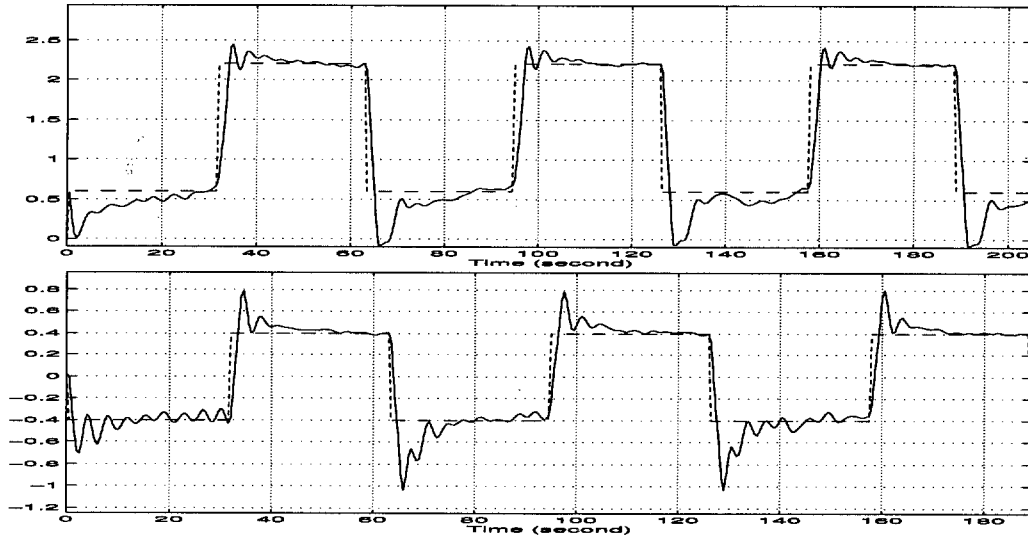


Fig. 9.25: Experimento con el controlador de estabilización y seguimiento: PID-RGO en variables de estado. Arriba: salida del sistema vs. referencia en eje vertical. Abajo: salida del sistema vs. referencia en eje horizontal.

Podemos además aprovechar las características convenientes de la convergencia global si al principio tomamos una bola lo suficientemente grande, de manera que reduzcamos la probabilidad de quedarnos en un mínimo local. Una vez que sepamos que seguimos al mínimo adecuado, podemos reducir el tamaño de la bola y al mismo tiempo de la población, para conseguir una mayor velocidad de ejecución del algoritmo.

Para que el algoritmo no tenga una convergencia prematura y se homogeneice demasiado la población, podemos aumentar el parámetro de probabilidad de mutación.

9.8 Comparación de los distintos métodos

Como se puede ver en la tabla, al introducir el control predictivo y el PID-RGO adaptativo se han mejorado los resultados, como era de prever.

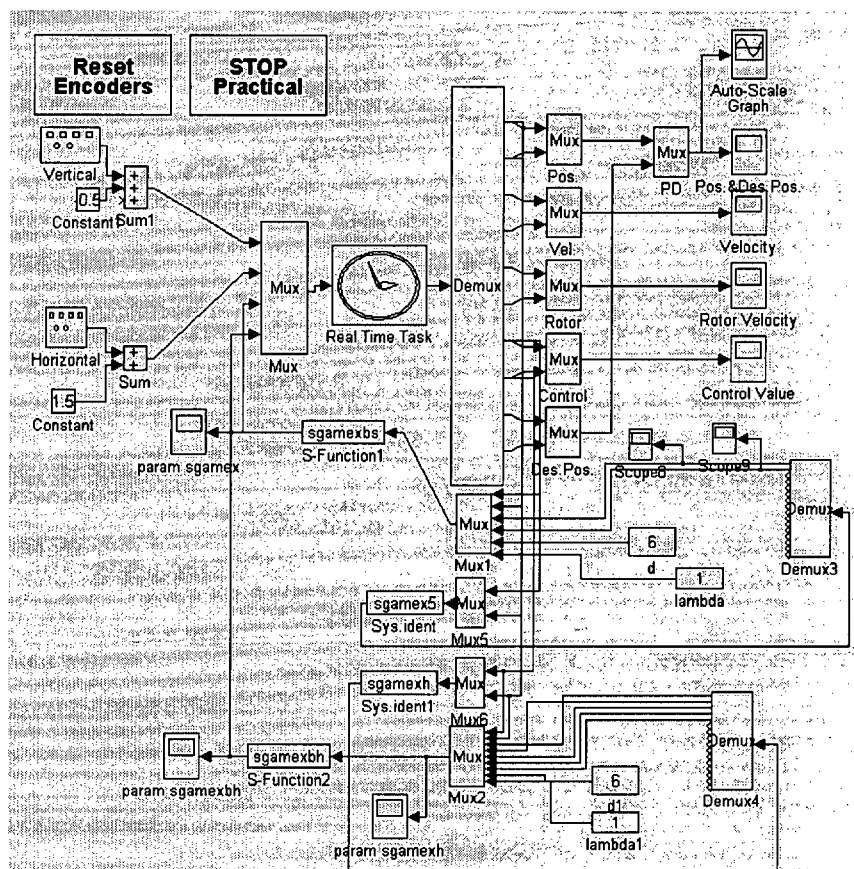


Fig. 9.26: Experimento con el controlador de estabilización: diagrama de bloques.

El control predictivo no se pudo introducir en el eje horizontal porque el tiempo de cómputo era mayor que la constante de tiempo del sistema y no funcionaba bien el controlador.

En resumen el método RGO se ha podido aplicar al control del TRMS con buenos resultados. El método ha demostrado ser fiable y con buen comportamiento con el único inconveniente de que el tiempo de cómputo es grande y esto impide aplicarlo a sistemas muy rápidos.

Tab. 9.2: Resultados de la comparación entre los distintos métodos de control del TRMS.

Error RMS		
	Plano vertical	Plano Horizontal
PID Vert. (1 DOF)	0.3980	—
PID Horiz. (1 DOF)	—	0.5520
PID (2 DOF)	0.4260	1.0326
PID con acoplamiento cruzado (2 DOF)	0.4284	1.0206
Predictivo-RGO Vertical	0.1982	—
PID-RGO Horizontal	—	0.2145
Predictivo-RGO y PID-RGO(2 DOF)	0.2038	0.2181
SS-PID-RGO (2 DOF)	0.2111	0.2212

10. APLICACIONES II: ROBOT MÓVIL

La localización de los robots móviles se basa en la odometría. El problema que presenta este método es que se van acumulando los errores y por lo tanto la precisión es cada vez menor. Para resolver esta situación se utilizan normalmente dos métodos: fundir la información sensorial con el mapa del entorno y el otro método basado en la detección de marcas naturales o artificiales. El método que nosotros hemos utilizado es el de detección de marcas artificiales para la localización del robot, empleando para ello como sensores una cámara con zoom motorizado sobre una plataforma 'pan and tilt' y un anillo periférico de 24 sensores periféricos. Para fusionar toda la información y así poder corregir la posición y orientación del vehículo utilizamos un filtro de Kalman extendido.

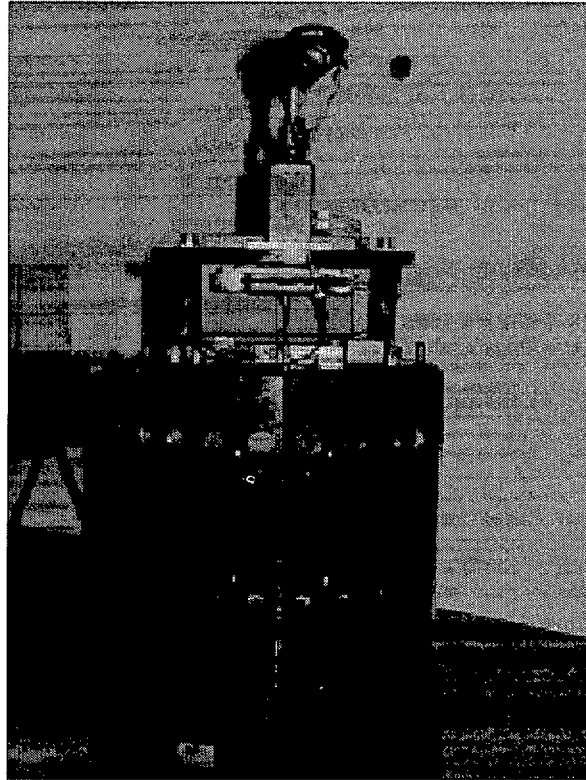
El filtro de Kalman extendido que se ha utilizado emplea la técnica de Restricted Genetic Optimization (RGO) para mejorar el funcionamiento del filtro de Kalman.

10.1 El Filtro de Kalman Extendido Iterado

Cuando el estado de un sistema tiene que ser estimado a partir de información sensorial con ruido, se necesita emplear algún tipo de estimador de estados para fusionar los datos de los diferentes sensores para producir una estimación precisa del verdadero estado del sistema. Si la dinámica del sistema y el modelo de observación son lineales, entonces el estimador de mínimos cuadrados puede ser calculado usando el *filtro de Kalman*. Cuando esto no es posible es necesario utilizar un filtro extendido, como por ejemplo el Filtro Iterado de Kalman [ver cap. 6 para más detalles].

10.2 Localización del Robot Móvil utilizando un Filtro RGO No-lineal

Los algoritmos genéticos son un proceso probabilístico de búsqueda basado en la selección natural y en las leyes genéticas. La población $\mathcal{J} = (j_1, j_2, \dots, j_n) \in J^N$



es modificada de acuerdo con los procesos evolutivos naturales: después de la inicialización, son ejecutados la selección $\omega : J^N \rightarrow J^N$, cruce $\chi : J^N \rightarrow J^N$, y mutación $\Xi : J^N \rightarrow J^N$ recursivamente en un bucle. Cada ejecución del bucle es llamada *generación* y J representa la población en la generación τ .

El operador de selección tiene como fin mejorar la calidad media de la población dando a los individuos de mayor calidad una alta probabilidad de ser copiados en la nueva generación. La selección, por lo tanto, enfoca la búsqueda hacia regiones prometedoras del espacio de búsqueda. La calidad de un individuo es medida por la función de salud $f : J \rightarrow \mathbb{R}$, donde J representa el espacio de todos los posibles individuos.

Los algoritmos genéticos se utilizan usualmente como método de optimización global de funciones invariantes en el tiempo, y usualmente se ejecutan 'off line'. Sin embargo, la selección natural de los seres vivos es un proceso local o semi local en el que las especies se adaptan por sí mismas al entorno, que a su vez es dependiente del tiempo (es 'on line').

Es posible adaptar el método de algoritmos genéticos si se restringe la bús-

queda a un entorno de la estimación previa usando como función de salud: $f : B(\hat{x}(k|k-1), \sigma) \rightarrow \mathbb{R}$, with $\sigma = \|P(k|k-1)\|$,

$$f(J) = \frac{1}{10^{-8} + V(\hat{x}(k|k-1))} \quad (10.1)$$

and

$$V(x(k)) = \frac{1}{2}(z(k) - h(k, x(k)))'R(k)^{-1}(z(k) - h(k, x(k))) + \frac{1}{2}(x(k) - \hat{x}(k|k-1))'(P(k|k-1))^{-1}(x(k) - \hat{x}(k|k-1)) \quad (10.2)$$

Por lo tanto el algoritmo actualiza las estimaciones de los estados y la matriz de covarianza y el bucle es repetido de nuevo, como se ilustra en la figura 1. La función de densidad de probabilidad de $x(k)$ dado Z^k viene dada como

$$\begin{aligned} p(x(k)|Z^k) &= p(x(k)|z(k), Z^{k-1}) = \\ &= \frac{1}{c} p(z(k)|x(k)) p(x(k)|Z^{k-1}) = \\ &= \frac{1}{c} N(z(k); h(k, x(k)), R(k)) N(x(k); \hat{x}(k|k-1), P(k|k-1)) \end{aligned} \quad (10.3)$$

Maximizar la función anterior es equivalente a calcular una estimación maximum a posteriori (MAP). Esto es también equivalente a minimizar $V(x(k))$, i.e. maximizar la función de salud $f(J)$.

La función de salud estándar (i.e. dividida por la suma de las saludes) es una aproximación de la función de densidad condicional (PDF).

$$p(x(k)|Z^k) = \frac{p(z(k)|x(k))p(x(k)|Z^{k-1})}{\int p(z(k)|x(k))p(x(k)|Z^{k-1})} \quad (10.4)$$

De todo lo anterior queda claro que es posible calcular precisamente las no linealidades de las funciones f y g , sin embargo la introducción de la hipótesis de ruido gaussiano no se puede evitar. Para determinar el radio de la zona de búsqueda, usamos la distancia de Mahalanobis

$$d = (\hat{x}(k|k-1) - \hat{x}(k-1|k-1))'P^{-1}(k|k)(\hat{x}(k|k-1) - \hat{x}(k-1|k-1)) \quad (10.5)$$

que mide la incertidumbre de la estimación $\hat{x}(k)$.

10.2.1 Consideraciones de diseño

La arquitectura del vehículo está basada en una arquitectura híbrida (AFREB) con un amplio rango de metodologías reactivas de control. La arquitectura AFREB

consiste en los siguientes módulos: supervisor de fusión, primitivas de comportamiento y ejecutor. La función del módulo supervisor de fusión es el cálculo del peso asignado a cada primitiva de comportamiento. La función del módulo ejecutor es el cálculo de las ordenes dadas al robot basadas en los pesos de las primitivas de comportamiento. Las entradas estándar del ejecutor son la velocidad v y la curvatura ω . La primitiva de comportamiento puede ser caracterizada por una secuencia temporal de valores apropiados para v y ω que causan que el robot exhiba una primitiva de comportamiento pre-especificada.

10.2.2 La estructura genética

Para resolver el problema de localización del robot móvil por medio de RGO, necesitamos un esquema de codificación para codificar los parámetros del problema en cadenas genéticas. Aquí, la posición del robot se codifica como una cadena por sus Coordenadas Cartesianas como:

$$\{(x_1, y_1, \theta_1), (x_2, y_2, \theta_2), \dots, (x_n, y_n, \theta_n)\} \quad (10.6)$$

con todos los valores guardados en forma decimal. Un conjunto de posiciones válidas aleatorias es creado como generación inicial alrededor de la solución dada por el filtro extendido de Kalman. Para resolver el problema, definimos un cromosoma con una longitud de 11 bits y poblaciones con un tamaño de 250 individuos. Después de que una nueva generación ha sido creada, calculamos la función de salud de cada individuo (posición). Definir funciones de salud apropiadas es vital para conseguir buenos rendimientos en la optimización.

10.2.3 La función de salud

Para optimizar la estructura utilizando RGO, uno debe ser capaz de asignar una medida de la calidad de cada estructura en el espacio de búsqueda. La función de salud es responsable de esta tarea. La función de salud utilizada puede ser definida de la siguiente manera:

- Marcas de Referencia:

$$g_l^k = \frac{1}{(\theta_{l-pan}^{k-e} - \theta_{pan}^{ob})^2 + (\theta_{l-tilt}^{k-e} - \theta_{tilt}^{ob})^2} \quad (10.7)$$

donde g_l^k es el coste de la posición correspondiente a la l -ésima cadena de la k -ésima generación, θ_{l-pan}^{k-e} , θ_{l-tilt}^{k-e} son los ángulos estimados entre el robot

y la marca de referencia de la l -ésima posición en la población en la k -ésima generación, respectivamente.

$$\theta_{l-pan}^{k-e} = \arcsin\left(\frac{y_a - p_y}{\sqrt{(x_a - p_x)^2 + (y_a - p_y)^2}}\right) \quad (10.8)$$

$$\theta_{l-tilt}^{k-e} = \arctan\left(\frac{p_z - z_a}{\sqrt{(x_a - p_x)^2 + (y_a - p_y)^2}}\right) \quad (10.9)$$

donde (p_x, p_y, p_z) es la posición de la marca de referencia y (x_a, y_a, z_a) es la posición de la plataforma pan-tilt. La detección normalizada en escala de grises se utiliza para la detección de marcas. La operación de correlación puede ser vista como una forma de convolución donde el encaje con el modelo patrón es análogo con el kernel de la convolución:

$$r(x, y) = I(x, y) \circ M(x, y) = \sum_{i=0}^{i=N} \sum_{j=0}^{j=N} M^*(i, j) I(x+i, y+j) \quad (10.10)$$

es decir, cada resultado de N pixels del modelo $M(x, y)$ es multiplicado por los N pixels de la imagen base $I(x, y)$, y todos estos productos son sumados. Finalmente el resultado es convertido en porcentaje, donde 100% significa que encaja perfectamente:

$$score = \max(r, 0)^2 * 100\% \quad (10.11)$$

Para separar los resultados de la correlación normalizada se considera un umbral de aceptación. Si la correspondencia entre la imagen y el modelo es menor que este nivel no se considera.

Los ángulos 'pan' y 'tilt' de la cámara respecto de la marca de referencia son:

$$\theta_{pan}^{ob} = \arctan \frac{d_{px}(C_x - u_c)}{f} \quad (10.12)$$

$$\theta_{tilt}^{ob} = \arctan \frac{d_{py}(v_c - C_y)}{f} \quad (10.13)$$

donde f es la distancia focal, (u_c, v_c) es el centro de la marca de referencia, (C_x, C_y) es el centro de la imagen y (d_{px}, d_{py}) es el tamaño de celda.

- Segmentos del entorno:

$$g_l^k = \frac{1}{\sum_{i=0}^n (D_{l-s}^{k-e} - D_s^{ob})^2} \tag{10.14}$$

donde g_l^k es el coste de la posición correspondiente a la l-ésima cadena de la k-ésima generación. D_{l-s}^{k-e} es la distancia estimada entre el sonar y el segmento de la l-ésima generación, respectivamente.

$$D_{l-s}^{k-e}(i) = \frac{1}{(\cos \theta_s^n + \sin \theta_s^n \tan \theta_s^n)} \frac{X_r - X_s^n}{(Y_s^n - Y_r) \tan \theta_s^n - \sin(\theta_{s-u}^i + \theta_r - \theta_s^n)} \tag{10.15}$$

donde (X_r, Y_r, θ_r) es la posición y orientación del vehículo, $(X_s^n, Y_s^n, \theta_s^n)$ es la posición y orientación del segmento observado y θ_{s-u}^a el ángulo del sensor.

10.3 Resultados experimentales

Con la finalidad de comprobar la fiabilidad y robustez del sistema de relocalización se ha realizado un estudio comparativo, con el vehículo detenido, con la finalidad de evaluar la mejora que supone la introducción del método anteriormente descrito en el proceso de localización del vehículo.

Los resultados obtenidos se muestran en la siguiente tabla:

	Núm. generaciones		
	2ª gen.	3ª gen.	4ª gen.
Con RGO	40	12	8
Sin RGO	60	31	29

Tab. 10.1: Localización en posición con y sin RGO (1). Error en mm.

En las tablas 10.1 y 10.2 se muestra la evolución del error de posición y orientación del vehículo, al introducir en el sistema errores respectivos de 20 cm y 10deg.

En las tablas 10.3 y 10.4 se han introducido simultáneamente errores de posición y ángulo.

	Núm. generaciones		
	2ªgen.	3ªgen.	4ªgen.
Con RGO	-2.6	1.8	0.2
Sin RGO	-3.9	2.6	-2

Tab. 10.2: Localización angular con y sin RGO (1). Error en grados.

	Núm. generaciones		
	2ªgen.	3ªgen.	4ªgen.
Con RGO	49	11	8
Sin RGO	61	31	18

Tab. 10.3: Localización en posición con y sin RGO (2). Error en mm.

	Núm. generaciones		
	2ªgen.	3ªgen.	4ªgen.
Con RGO	3.7	-0.5	1
Sin RGO	4.1	-5.2	2.1

Tab. 10.4: Localización angular con y sin RGO (2). Error en grados.

En las tablas anteriores, se pone de manifiesto la mejora que produce la implantación del método descrito en el proceso de localización. El principal inconveniente radica en el elevado tiempo de ciclo que requiere para su ejecución, del orden de 15 veces el necesario para el método de la proyección.

11. CONCLUSIONES Y TRABAJO FUTURO.

11.1 Conclusiones

11.1.1 Resumen y conclusiones

En esta tesis se ha desarrollado un nuevo método 'Optimización Genética Restringida' (RGO) y las ventajas de su utilización en la identificación de sistemas no lineales, en la estimación de estados (Filtro de Kalman extendido), en el diseño de controladores adaptativos y aplicaciones a la relocalización de Robots móviles..

11.2 Aportaciones principales

Se ha desarrollado el método de optimización RGO (Restricted Genetic Optimization) para funciones que varíen con el tiempo. Este método ha probado ser muy útil para minimizar o maximizar funciones no-lineales y no-diferenciables. Este método está especialmente indicado para su uso en la identificación para el control de sistemas dinámicos.

El método RGO ha sido utilizado en la identificación en la forma de entrada-salida [cap. 5], y en la forma de variables de estado [cap. 7] con excelentes resultados. Las ventajas principales han sido su utilización 'en línea' y su aplicabilidad a sistemas no-lineales.

También ha sido aplicado este método a la estimación de estados [cap 6], creando dos nuevos filtros de Kalman extendidos, que se pueden aplicar a la estimación de estados 'en línea', de sistemas no lineales y los resultados han sido muy buenos. Este método ha sido aplicado muy eficientemente a la localización de robots móviles [cap 10].

Seguidamente, se ha aplicado a la realización de controladores adaptativos no lineales basados en el método RGO: PID-RGO[cap. 8, 9], controlador predictivo RGO [cap. 8, 9] y controlador en variables de estado RGO [cap. 9].

Estos controladores han demostrado su valía al ser aplicados al control del TRMS (Twin Rotor MIMO System)[cap 9], que es un laboratorio diseñado expresamente para realizar experimentos de control y en ciertos aspectos su comportamiento recuerda el de un helicóptero. Desde el punto de vista de control ejemplifica un sistema de orden alto con acoplamientos cruzados muy significativos.

En el

11.3 Trabajo Futuro

El método RGO ha demostrado ser muy efectivo en resolver problemas de identificación y control.

En el futuro, será muy interesante aplicarlo al reconocimiento de la estructura del modelo y la identificación de qué regresores son los más significativos en la construcción de dicho modelo.

También estamos pensando aplicarlo a diversas partes de la robótica como la localización de robots y la planificación de trayectorias.

Una tercera línea de avance es la aplicación de otros algoritmos evolutivos, tales como las Estrategias Genéticas (Evolutionary Strategies) y la Programación Genética (Genetic Programming) para manejar estos problemas y comparar los resultados. Los Algoritmos Genéticos Paralelos (Parallel Genetic Algorithms) también pueden ser de gran ayuda a la hora de mejorar las prestaciones de los algoritmos diseñados y con más aplicaciones en el mundo real.

BIBLIOGRAFÍA

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19:716–723, 1974.
- [2] H. Akaike. Markovian representation of sthochastic processes by canonical variables. *SIAM Journal of Control and Optimization*, 13:162–173, 1975.
- [3] J. M. Armingol. *Localización Geométrica de Robots Móviles Autónomos*. PhD thesis, Univ. Carlos III, Madrid, Spain, 1997.
- [4] J. M. Armingol, L. E. Moreno, S. Garrido, A. de la Escalera, and M. A. Salichs. Mobile robot localization using a non-linear evolutionary filter. In *3rd IFAC Symp. on Intelligent Autonomous Vehicles (IAV'98)*. IFAC, 1998.
- [5] K. Åström and T. Hägglund. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, Research Triangle Park, N. C., 1995.
- [6] K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.
- [7] K. J. Åström and B. Wittenmark. *Computer Controlled Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [8] K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, Reading, Massachusetts, 1995.
- [9] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. The Academic Press, London, 1988.
- [10] Y. Bar-Shalom and Xiao-Rong. *Estimation and Tracking*. Artech House, 1993.
- [11] C. Bordóns. *Control Predictivo Generalizado de Procesos Industriales: Formulaciones Aproximadas*. PhD thesis, ETSII, Univ. de Sevilla, Spain, 1994.



- [12] G. E. P. Box and D. R. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco, 1980.
- [13] E. F. Camacho and C. Bordóns. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, 1995.
- [14] C. Canudas. *Adaptive Control for Partially Known Systems*. Elsevier, Amsterdam, 1988.
- [15] R. J. Carroll and D. Ruppert. *Transformation and Weighting in Regression*. Chapman and Hall, New York, 1988.
- [16] A. Chipperfield and P. Fleming. Genetic algorithms in control systems engineering. *Control and Computers*, 23(3):88–94, 1995.
- [17] C. T. Chou. *Geometry of Linear Systems and Identification*. PhD thesis, Trinity College, Cambridge, England, 1994.
- [18] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized Predictive Control: Part I. The Basic Algorithm. *Automatica*, 23(2):137–148, 1987.
- [19] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized Predictive Control: Part II. Extensions and Interpretations. *Automatica*, 23(2):149–160, 1987.
- [20] A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Trans. on Mathematical Software*, 13(3), 1987.
- [21] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.
- [22] A. Pérez de Madrid. *Aplicación de Las Técnicas de Programación Dinámica al Control Predictivo Basado En Modelos*. PhD thesis, Facultad de Ciencias, UNED, Madrid, Spain, 1995.
- [23] P. M. J. Van den Hof and R. J. P. Schrama. Identification and control – closed-loop issues. *Automatica*, 31(12):1751–1770, 1995.
- [24] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

- [25] S. J. Flockton and M. J. White. Pole-zero system identification using genetic algorithms. In *Proceedings 5th International Conference on Genetic Algorithms*, pages 531–535. University of Illinois at Urbana Champaign, USA, 17-21 July 1993.
- [26] U. Forsell and L. Ljung. Closed-Loop Identification Revisited. Technical report, Linköping University, Linköping, Sweden, 1997.
- [27] U. Forsell. *Closed-loop Identification*. PhD thesis, Linköping University, Linköping, Sweden, 1999.
- [28] U. Forsell and L. Ljung. Issues in closed-loop identification. Technical report, Linköping University, Linköping, Sweden, 1998.
- [29] S. Garrido, L. E. Moreno, and C. Balaguer. State estimation for nonlinear systems using restricted genetic optimization. In *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. IEA-AIE, 1998.
- [30] S. Garrido, L. E. Moreno, and M. A. Salichs. Nonlinear on-line identification of dynamic systems with restricted genetic optimization. In *6th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, 1998. EUFIT.
- [31] S. Garrido, L. E. Moreno, and M. A. Salichs. Identification of state space models with rgo. In *7th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, 1999. EUFIT.
- [32] P. J. Gawthrop. Parametric identification of transient signals. *IMA Journal of Mathematical Control and Information*, 1:117–128, 1984.
- [33] W. Gesing and E.J. Davison. An exact penalty function algorithm for solving general constrained parameter optimization problem. *Automatica*, 15:175–188, 1979.
- [34] M. Gevers. Towards a joint design of identification and control. In H. L. Trentelman and J. C. Willems, editors, *Essays on Control: Perspectives in the Theory and its applications*, pages 111–151. Birkhäuser, 1993.
- [35] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1993.

- [36] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 1989.
- [37] G. C. Goodwin and K. S. Sin. *Adaptive Filtering, Prediction and Control*. Prentice-Hall, Englewood Cliffs, N. J., 1984.
- [38] G. J. Gray and D. J. Murray-Smith. Nonlinear model structure identification using genetic programming. *Control Eng. Practice*, 6(11):1341–1353, 1999.
- [39] U. Grenander. *Abstract Inference*. Wiley, New York, 1981.
- [40] I. Gujon and P. Albrecht. A time delay neural network character recognizer for touch terminal. Technical report, ATT Bell Laboratories, 1990.
- [41] I. Gustafsson, L. Ljung, and T. Söderstrom. Identification of processes in closed loop – identifiability and accuracy aspects. *Automatica*, 13:59–75, 1977.
- [42] R. L. Haupt and S. E. Haupt. *Practical Genetic Algorithms*. Wiley-Interscience, New York, 1998.
- [43] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 350 Bridge Parkway, Redwood City, CA 94065, 1991.
- [44] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- [45] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimizations problems. *Biol. Cybern*, 1986.
- [46] T. C. Hsia. *System Identification: Least Squares method*. Lexington Books, 1977.
- [47] K. Hunt, D. Sbarbaro, R. Zbikowski, and P. Gathrop. Neural networks for control systems - a survey. *Automatica*, 28(6):1083–1112, 1992.
- [48] H. Hyötyniemi. Regularization of parameter estimation. In *13th IFAC World Congress*, San Francisco, California, 1996.
- [49] H. Hyötyniemi. On Structural Identifiability of Dynamic Models. Technical report, Helsinki University of Technology, Helsinki, Finland, 1998.

- [50] H. Iba, H. deGaris, and T. Sato. A numerical approach to genetic programming for system identification. *Evolutionary Computation*, 28(6):1083–1112, 1995.
- [51] R. Isermann. Practical aspects of process identification. *Automatica*, 16:575–587, 1980.
- [52] R. Isermann. *Digital Control Systems. Volume I. Fundamentals, Deterministic Control*. Springer-Verlag, Berlin, 1989.
- [53] R. Isermann. *Digital Control Systems. Volume II. Stochastic Control, Multivariable Control, Adaptive Control, Applications*. Springer-Verlag, Berlin, 1989.
- [54] R. Isermann, K.H. Lachmann, and D. Matko. *Adaptive Control Systems*. Prentice-Hall International, Hemel Hempstead, UK, 1992.
- [55] R. Johansson. *System Modeling and Identification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
- [56] C. R. Johnson. Admissibility in blind adaptive channel equalization. *IEEE Control Systems Magazine*, 1991.
- [57] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, 12(31):1725–1750, 1995.
- [58] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. a new approach for the nonlinear transformation of means and covariances in linear filters. *IEEE Trans. on Automatic Control*, 1996.
- [59] P. T. Kabamba. Balanced forms: Canonicity and parametrization. *IEEE Trans. on Automatic Control*, 30(11):1106–1109, 1985.
- [60] T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
- [61] R. E. Kalman. Identifiability and modelling in econometrics. In P. R. Krishnaiah, editor, *Developments in Statistics*, volume 4. Academic Press, London, 1983.

- [62] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng. ASME*, Series D:95–108, 1961.
- [63] G. Kitagawa. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.
- [64] K. Kristinsson and G. Dumont. System identification and control using genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):1033–1046, 1992.
- [65] S. Kung. *Digital Neural Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
- [66] C. Ledoux. Identification of SISO Nonlinear Wiener Systems. Technical report, Linköping University, Linköping, Sweden, 1996.
- [67] W. S. Lee, B. D. O. Anderson, I. M. Y. Mareels, and R. L. Kosut. On some key issues in the windsurfer approach to adaptive robust control. *Automatica*, 31(11):1619–1636, 1995.
- [68] L. Ljung. *System Identification Toolbox - User's Guide*. The Mathworks, 1986.
- [69] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, N. J., 1987.
- [70] L. Ljung. Identification for Control - What Is There To Learn? Technical report, Linköping University, Linköping, Sweden, 1998.
- [71] L. Ljung and S. Gunnarsson. Adaptation and tracking in system identification - a survey. *Automatica*, 26:7–21, 1990.
- [72] L. Ljung and T. McKelvey. A Least Squares Interpretation of Sub-Space Methods for System Identification. Technical report, Linköping University, Linköping, Sweden, 1996.
- [73] L. Ljung, Sjöberg, and T. McKelvey. On the Use of Regularization in System Identification. Technical report, Linköping University, Linköping, Sweden, 1992.
- [74] L. Ljung and T. Söderstrom. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, Massachusetts, 1983.

- [75] W. Luo and S. A. Billings. Adaptive model selection and estimation for nonlinear systems using a sliding data window. *Signal Processing*, 46:179–202, 1995.
- [76] J. M. Maciejowski. Balanced realizations in system identification. In *7th IFAC Symposium on Identification and parameter Estimation*, York, UK, 1985.
- [77] J. M. Maciejowski. *Multivariable Feedback Design*. Addison-Wesley, Wokingham, England, 1989.
- [78] P. S. Maybeck. *Stochastic models, Estimation, and Control*. Academic Press, London, 1982.
- [79] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, New York, 1989.
- [80] T. McKelvey. *Identification of State-Space Models from Time and Frequency Data*. PhD thesis, Linköping University, Linköping, Sweden, 1995.
- [81] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. AI Series. Springer Verlag, New-York, 1994.
- [82] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. on Automatic Control*, 26(1):17–32, 1981.
- [83] K. Narendran and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1:4–27, 1990.
- [84] K. Narendran and K. Parthasarathy. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans on Neural Networks*, 2(2):252–262, 1991.
- [85] J. P. Norton. *An Introduction to Identification*. Academic Press, London, 1986.
- [86] R. J. Ober. Balanced realizations: Canonical form, parametrization, model reduction. *Int. J. Control*, 46(2):643–670, 1987.
- [87] M. Ouladsine, F. Bicking, and G. Bloch. Identification of constrained dynamic systems by genetic type algorithm. In *IFAC Artificial Intelligence in Real-Time Control*, Blend, Slovenia, 1995.

- [88] A. J. M. Van Overbeek and L. Ljung. On-line structure selection for multi-variable state space models. *Automatica*, 18(5):529–543, 1992.
- [89] P. Poddar and K. P. Unnikrishnan. Memory neuron networks: a prolegomenon. Technical Report GMR-7493, G. M. Research Laboratories, Warren, Michigan, 1991.
- [90] E. Polak and D. Q. Mayne. An algorithm for optimization problems with functional inequality constraints. *IEEE Trans. on Automatic Control*, 21:184–193, 1976.
- [91] B. Porter and A. H. Jones. Genetic tuning of digital pid controllers. *Electronic Letters*, 28:843–844, 1992.
- [92] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. part i: Basic properties of selection and mutation. *IEEE Trans. on Neural Networks*, 5(1):102–119, 1994.
- [93] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. part ii: Analysis of the diversification role of crossover. *IEEE Trans. on Neural Networks*, 5(1):120–129, 1994.
- [94] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Parallel Distributed Processing*, volume 1, chapter Learning internal representations by error backpropagation, pages 318–362. MIT Press, 1986.
- [95] J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural Networks in System Identification. Technical report, Linköping University, Linköping, Sweden, 1996.
- [96] J. Sjöberg, Q. Zhang, and L. Ljung. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, 1995.
- [97] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall International, Hemel Hempstead, Hertfordshire, 1989.
- [98] E. Sontag. *Essays on Control: Perspectives in the Theory and its Applications*, volume 14 of *Progress in Systems and Control Theory*, chapter Neural Networks for control, pages 339–380. Birkhäuser, 1990.

- [99] K. C. Tan, Y. Li, D. J. Murray-Smith, and K. C. Sharman. System identification and linearization using genetic algorithms with simulated annealing. In *Proc. 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield, UK., 1995.
- [100] H. Tanizaki. *Nonlinear Filters*. Springer, Berlin, 1996.
- [101] H. Tanizaki and R. S. Mariano. Nonlinear filters based on taylor series expansions. *Communications in Statistics, Theory and Methods*, 25(6), 1996.
- [102] R. Valverde. *Control de Sistemas mediante Redes Neuronales. Aprendizaje por Refuerzo*. PhD thesis, Univ. Carlos III, Madrid, Spain, 1999.
- [103] P.E. Wellstead and M.B. Zarrop. *Selftunning Systems: Control and Signal Processing*. John Wiley and Sons, Chichester, U.K., 1991.
- [104] P. J. Werbos. Back propagation through time: What it does, how do it. *Proceedings IEEE*, 1990.
- [105] D. White and D. Sofge, editors. *Handbook of Intelligent Control, Neural, Fuzzy, and Adaptive Approaches*. Multiscience Press, Inc., Van Nostrand Reinhold, 115 Fifth Avenue, New York, NY 10003, 1992.
- [106] R. P. Wishner, J. A. Tabaczynski, and M. Athans. A comparison of three non-linear filters. *Automatica*, 5:487–496, 1989.
- [107] Z. Yang, T. Hachino, and T. Tsuji. On-line identification of continuous time delay systems combining least squares techniques with genetic algorithms. *Int. J. Control*, 66(1):23–42, 1996.
- [108] D. Yuret. From Genetic Algorithms to Efficient Optimization. Technical report, MIT, Cambridge, Massachusetts, 1996.
- [109] L. A. Zadeh. From circuit theory to system theory. In *Proc. IRE 50*, pages 856–865, 1962.